Die vorliegende Abhandlung

Über die Anwendung, Klassifizierung und Übertragbarkeit von Methoden für einen Ähnlichkeitsabgleich von Geschäftsprozessmodellen

wurde am
07. Dezember 2017
von

Frau **Michaela Baumann,** M.Sc.,
geboren in Kempten (Allgäu),

der Universität Bayreuth
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
vorgelegt und ist von dieser genehmigt.

1. Gutachter: Prof. Dr.-Ing. Stefan Jablonski
2. Gutachter: Prof. Dr. Jörg Rambau

Tag des Kolloquiums: 19. April 2018

Kurzfassung

Der Einsatz von Prozessmodellen zur Abbildung und Umsetzung betrieblicher Arbeitsabläufe wird in immer mehr Unternehmen durchgesetzt. Hierbei wachsen die verwendeten Modellrepositorien schnell an, es muss also eine Vielzahl an Prozessmodellen effektiv verwaltet werden. Zu dieser Verwaltung zählt auch, Modellabgleiche durchzuführen, das heißt, gleiche oder ähnliche Prozessmodelle in der Masse an Modellen zu erkennen und Hinweise zu einer Verschmelzung solcher Modelle zu geben. Mehrfachmodellierungen und Modellvarianten können hierbei verschiedene Ursachen haben, die von Unternehmensfusion über zielgruppengerichtete Modellierung bis hin zu Modellevolution reichen. Auch bei der Übersetzung von Modellen in eine andere Modelliersprache oder bei der Evaluation solcher Übersetzungsmethoden ist ein Abgleich von Modellen notwendig.

In der Literatur werden einige Möglichkeiten für Prozessmodellabgleiche vorgeschlagen, die jedoch vornehmlich nur imperativ modellierte Prozesse abdecken und dort auch nur einen Teil der eigentlich zur Verfügung stehenden Modellinformationen nutzen. Bei diesen in der Literatur verwendeten Methoden wird, nachdem eine Abbildung zwischen zwei zu vergleichenden Prozessmodellen festgelegt ist, die jeweils eine Aktivität des ersten Modells auf eine Aktivität des zweiten Modells abbildet, wobei keine Aktivität mehrfach in dieser Abbildung auftritt, auf Basis dieser Abbildung ein Ähnlichkeitswert errechnet. Zur Berechnung werden Informationen über die Struktur des Prozessmodells, über das Verhalten des Prozesses und über die Aktivitätsbeschreibungen genutzt. Der Ähnlichkeitswert ist eine Zahl zwischen null (keine Ähnlichkeit) und eins (volle Ähnlichkeit), der über die zugrunde liegende Abbildung maximiert wird.

Die vorliegende Arbeit erweitert die Ansätze aus der Literatur in mehreren Punkten. Zusätzlich zu den Aktivitätsbeschreibungen, der Struktur und dem modellierten Verhalten werden verwendete Datenobjekte, zuständige Agenten bzw. Rollen oder Personengruppen sowie benutzte Werkzeuge/Services zur Ähnlichkeitsberechnung herangezogen. Die Möglichkeiten der zugrunde liegenden Abbildung werden von einer 1:1-Abbildung zu einer M:N-Abbildung erweitert, das heißt, es können Mengen von Aktivitäten auf Mengen von Aktivitäten abgebildet werden, um unterschiedliche Granularitäten der zu vergleichenden Modelle zu berücksichtigen. Des Weiteren wird eine Ausweitung der Abgleichsansätze auf deklarativ modellierte Prozesse, das sind Prozesse, die auf einem Regelsystem beruhen, besprochen. Eine Proof of Concept-Implementierung einiger Abgleichsansätze sowie mehrere vergleichende Evaluationen zeigen die Funktionsweise und Anwendbarkeit der vorgestellten Methoden.

Abstract

The use of process models for the mapping and implementation of company workflows is being enforced in more and more companies. The model repositories used grow rapidly, thus, a large number of process models need to be managed effectively. This administration also includes the conduction of model matching, that is, identifying equal or similar process models in the mass of models and giving advice for merging such models. Multiply modeled process models and model variants can have various causes, ranging from company fusion to target group oriented modeling to model evolution. A comparison of models is also necessary when translating models into another modeling language or when evaluating such translation methods.

In the literature, some possibilities for process model matching are proposed, which, however, primarily cover only imperatively modeled processes and thereby use only a part of the actually available model information. In these methods used in the literature, after a mapping, which maps an activity of the first model to an activity of the second model, where no activity occurs twice in the mapping, between two process models to be compared is defined, a similarity value is computed based on the mapping. The calculation uses information about the structure of the process model, the behavior of the process, and the activity descriptions. The similarity value is a number between zero (no similarity) and one (full similarity) that is maximized across the underlying mapping.

The work at hand extends the approaches from the literature in several aspects. In addition to the activity descriptions, the structure, and the modeled behavior, data objects, responsible agents, roles or groups of persons, as well as utilized tools/services are used for the similarity calculation. The possibilities of the underlying mapping are extended from a 1:1 mapping to an M:N mapping, which means that sets of activities can be mapped to sets of activities in order to consider different granularities of the models to be compared. Furthermore, an extension of the matching approaches to declaratively modeled processes, i.e., processes represented as constraint-based models, is discussed. A proof of concept implementation of some matching approaches as well as several comparative evaluations show the functionality and applicability of the presented methods.

Inhaltsverzeichnis

Kapitel 1

Motivation

Der Einsatz von Geschäftsprozessmodellen in Unternehmen hat in den vergangenen Jahren stark zugenommen. Unter anderem zur Unterstützung von Dokumentationsaufgaben, was vom Qualitätsmanagement oder sogar vom Gesetzgeber vorgeschrieben sein kann, zur Gestaltung bzw. Umgestaltung tatsächlicher Arbeitsabläufe oder auch zur Implementierung von Informationssystemen werden solche Modelle herangezogen (Weske, 2010). Um dabei sämtliche Eigenheiten eines Prozesses zu erfassen und festzuhalten, werden typischerweise mehrere Modellierungsexperten sowie Domänenexperten aus unterschiedlichen Geschäftsbereichen in den Modellierungsprozess einbezogen (Dijkman, 2007). Dies, aber auch Umstände wie Firmenfusionen, kann dazu führen, dass eine Vielzahl an Prozessmodellen für ein und denselben Prozess vorhanden ist und ein anschließender Abgleich dieser Modelle durchgeführt werden muss (Dijkman et al., 2009b). Sich entsprechende Modellelemente, wie etwa gleiche Aktivitäten, aber auch Unterschiede zwischen den Modellen sollen dabei aufgezeigt werden. Um eine Einschätzung über den Grad an Ähnlichkeit zweier Modelle zu bekommen, wird zudem eine normierte Zahl, der Ähnlichkeitswert, ausgegeben. Zur Berechnung dieses Wertes werden die Beschreibung der Aufgaben, die Struktur der Modelle und andere Eigenschaften verwendet. Die Schwierigkeiten beim Abgleich liegen unter anderem darin, dass unterschiedliches Vokabular bei der Beschreibung der Aufgaben verwendet wird und dass der Detailgrad der Modelle stark verschieden sein kann. Außerdem können Modelle in imperativer oder in deklarativer Art und Weise formuliert sein, d. h. mittels fest vorgegebener Ausführungsmöglichkeiten oder mittels eines Regelwerks. Die bisher genannten Punkte – die Zielsetzung und die Schwierigkeiten des Ähnlichkeitsabgleichs sowie die Beschaffenheit der vorgegebenen Modelle – werden im Folgenden genauer erläutert, ehe der weitere Aufbau der Arbeit präsentiert wird.

1.1 Zielsetzung des Ähnlichkeitsabgleichs von Prozessmodellen

Der Vergleich von Prozessmodellen beispielsweise über eine Bestimmung ihrer Ähnlichkeit kann in ganz verschiedenen Bereichen Anwendung finden. Verwandte Arbeiten und eigene Erfahrungen lassen besonders vier Zielsetzungen eines Ähnlichkeitsabgleichs von Prozessmodellen erkennen:

- Das Auffinden von Duplikaten bzw. Varianten,
- das Prüfen von Konformität,
- die Verbesserung der Verständlichkeit sowie

- das Einsetzen als Evaluationsinstrument.

Diese Auflistung ist nicht erschöpfend, sie deckt aber ein breites Anwendungsfeld des Ähnlichkeitsabgleichs ab. In Becker und Laue (2012) ist eine Liste aus sieben Anwendungsfeldern für die Ähnlichkeitsberechnung von Prozessmodellen gegeben. Diese Anwendungsfelder sind größtenteils in Abschnitt 1.1.1 der vorliegenden Arbeit zusammengefasst.

1.1.1 Auffinden von Duplikaten bzw. Varianten

Das Entdecken von Duplikaten oder zumindest von Prozessvarianten kann für viele verschiedene Zwecke nützlich sein. Zum einen besteht die Notwendigkeit, große Prozessmodellrepositorien irgendwie zu verwalten. In solchen Repositorien können hunderte oder gar tausende von Modellen liegen (Rosemann, 2006). Um keine redundanten Modelle einzuführen (Weber et al., 2011) und somit ein unnötiges Anwachsen des Repositoriums zu vermeiden, was direkt mit dem Verwaltungsaufwand für dieses in Verbindung steht, ist das Erkennen von Duplikaten unbedingt notwendig. Dabei sollen sowohl exakte Klone als auch sehr ähnliche Varianten (Rosa et al., 2015; Ekanayake et al., 2012) entdeckt werden. Varianten von Prozessmodellen innerhalb eines Repositoriums können beispielsweise dann entstehen, wenn ähnliche Prozesse getrennt voneinander modelliert werden, ihre Ausführung aber mehr oder weniger dieselbe ist (Rosa et al., 2015), wenn ein Prozess sich nach und nach über die Jahre verändert hat und sein zugehöriges Modell stets entsprechend mit angepasst und neu gespeichert wurde, was zu vielen ähnlichen Modellen, also zu verschiedenen Versionen, führt (Zhao und Liu, 2007) oder wenn eine komplette Geschäftsprozessneugestaltung (*Business Process Reengineering*) (Tka und Ghannouchi, 2012) mit Auswirkungen auf viele vorhandene Prozessmodelle durchgeführt wird. Auch eine zielgruppenspezifische Modellanpassung im Rahmen der Internationalisierung (Dijkman et al., 2009b) oder ein wirtschaftliches Großereignis wie eine Unternehmensfusion, die auf einen Schlag etwa eine Verdopplung der Repositoriumsgröße erwarten lässt, kann zu mehrfach modellierten Prozessen führen. Die Erkennung von Varianten kann dann zu einem Zusammenführen der (unbeabsichtigten) Duplikate oder zu einer Clusterbildung der Varianten für ein einfacheres Verwalten des Repositoriums führen.

Das Wiederverwenden von ganzen Prozessmodellen oder von Teilen davon, beispielsweise von Subprozessen, ist ein anderer Nutzen des Ähnlichkeitsabgleichs, der darauf abzielt, dass das Modellieren an sich weniger Zeit benötigt und weniger kostenintensiv ist, da bereits modellierte Teilprozesse einfach erneut benutzt werden. Besonders im Zusammenhang mit Arbeitsabläufen im wissenschaftlichen Umfeld, zum Beispiel Abläufe von Experimenten, kann eine Wiederverwendung von bereits modellierten Abläufen hilfreich sein (Grigori et al., 2010). Auch bei der Integration von Web Services kann ein Ähnlichkeitsabgleich nutzen, wenn es darum geht, Services mit einem ähnlichen Verhalten, beschrieben über eine bestimmte Protokollsprache, ausfindig zu machen, um sie wiederverwenden zu können (Grigori et al., 2010). Von Lu und Gao (2008) wird das Modellieren an sich als weitere Einsatzmöglichkeit für die Ähnlichkeitsmessung von Prozessmodellen genannt. Fertigungsprozesse in der Industrie sollen mit Hilfe von Modellähnlichkeiten einfacher angepasst werden können um so Zeit und Kosten zu sparen.

1.1.2 Prüfen von Konformität

Manchmal ist es notwendig, dass Prozessmodelle einem bestimmten Referenzmodell entsprechen oder einer gegebenen Menge an Bedingungen genügen müssen. Dies kann gesetzliche

Gründe haben oder beispielsweise von der Abteilung für Qualitätssicherung in einem Unternehmen vorgegeben sein. Bei gesetzlichen Vorgaben muss der Abgleich natürlich exakt sein und alle vorgegebenen Bedingungen müssen vom untersuchten Prozessmodell erfüllt werden. Abweichungen des Prozessmodells vom Referenzmodell sollten sorgfältig geprüft werden, was voraussetzt, dass beim Ähnlichkeitsabgleich auch Abweichungen entdeckt werden müssen.

Awad et al. (2008) beschreiben die Notwendigkeit, Prozessmodelle in einem Respositorium ausfindig zu machen, die einer vorgegebenen Suchanfrage entsprechen. Von Reichert und Weber (2012) werden klinische Prozesse genannt, bei denen unter anderem aufgrund des Zusammenwirkens verschiedener Behandlungen oder Medikamente bestimmte Regeln, die über ein einzelnes Prozessmodell bzw. einen einzelnen Prozess hinausgehen, zum Wohle des Patienten immer eingehalten werden müssen.

1.1.3 Verbessern der Verständlichkeit

In der jüngeren Vergangenheit ist eine Reihe von verschiedensten Prozessmodellierungssprachen entstanden. Jede dieser Sprachen hat eine andere Ausrichtung, einen anderen Umfang und stellt eine gewisse Ausdrucksmächtigkeit zur Verfügung. Es dürfte nicht verwunderlich sein, dass man nicht all diese Sprachen beherrschen kann. Besonders Modelle, die deklarativ modelliert sind (siehe Abschnitt 1.3.3), sind nicht einfach zu verstehen, was aber in ihrer Natur liegt, da sie beispielsweise viele versteckte Abhängigkeiten beinhalten (Fahland et al., 2009a). Deswegen kann es hilfreich sein, ein Modell mit einem Modell, das in einer vertrauten Sprache modelliert ist, zu vergleichen und so Ähnlichkeiten und auch Unterschiede der beiden Modelle aufzuzeigen (Ackermann, 2017).

1.1.4 Einsetzen als Evaluationsinstrument

Der Einsatz von Prozessmodellabgleichen bietet sich auch dann an, wenn beispielsweise Transformationstechniken oder Miningalgorithmen evaluiert werden sollen. Aiolli et al. (2012) benötigen einen Abgleich von Prozessmodellen, um geeignete Parameter für Miningalgorithmen wählen zu können. Auch bei der Übersetzung von Modellen einer Modelliersprache in eine andere (Ackermann et al., 2017b) können die Modelle nach der Transformation verglichen werden, um die Güte der Übersetzung zu bestimmen.

1.2 Herausforderungen beim Bestimmen von Ähnlichkeiten

Bei der Bestimmung der Ähnlichkeit zweier Prozessmodelle gibt es auf der einen Seite Schwierigkeiten, die von außen gegeben sind; hauptsächlich ist dies die große Vielfalt an unterschiedlichen Prozessmodellierungssprachen. Grundsätzlich kann bei den Sprachen zwischen imperativen und deklarativen Prozessmodellierungssprachen unterschieden werden (vgl. Abschnitt 1.3). Doch auch innerhalb dieser beiden Klassen existieren unterschiedliche Sprachen. Zu diesen äußerlichen Schwierigkeiten kommen modellierungsinhärente Herausforderungen. Da ist zum einen der grundsätzliche Modellentwurf, also die Granularität (Feinheit), mit der ein Prozess modelliert wird, zum anderen die Umsetzung des Entwurfs zu nennen. Meist gibt es verschiedene Möglichkeiten, ein bestimmtes Verhalten darzustellen, außerdem kann, was Beschriftungen angeht, eine unterschiedliche Terminologie gewählt werden und innerhalb einer Terminologie auch auf verschiedene Arten formuliert werden (Weidlich et al., 2010a).

Eine wichtige Frage, die im Zusammenhang mit dem Ähnlichkeitsabgleich von Prozessmodellen auftritt, ist folgende: Wo ist der Grund bzw. sind die Gründe für eine Abweichung der

verglichenen Modelle? Das bedeutet, dass nicht nur ein Ähnlichkeitsmaß berechnet werden soll, es soll vielmehr auch spezifiziert werden können, wo Unterschiede zwischen den abgeglichenen Prozessmodellen liegen.

1.3 Prozesse und ihre Modellierung

Bevor mit der Modellierung eines Prozesses begonnen wird, sollte zunächst geklärt sein, um welche Art von Prozess es sich handelt. Es wird grundsätzlich zwischen Routineprozessen und agilen Prozessen unterschieden (Reichert und Weber, 2012). Aus der Art eines Prozesses leiten sich unterschiedliche Anforderungen an seine Modellierung ab, die entweder imperativ für Routineprozesse oder deklarativ für agile Prozesse sein kann. Zur Klärung der Begriffe soll hier noch der Unterschied zwischen „(Geschäfts-)Prozess“, „Prozessmodell (oder auch nur „Modell“ genannt) und „Prozessinstanz“ erläutert werden. Ein (Geschäfts-)Prozess ist eine Menge bestimmter Einzeltätigkeiten, die miteinander strukturiert zu Arbeitsabläufen verknüpft sind, um ein bestimmtes Ziel bzw. eine von einem Kunden erwartete Leistung zu erreichen (Gaitanides, 2012). Ein Prozessmodell bildet einen Prozess, also die Ablauforganisation, eines Unternehmens oder mehrerer Unternehmen ab und erfüllt dabei die von Stachowiak (1973) allgemein geforderten Bedingungen für ein Modell: das Abbildungsmerkmal (ein Modell repräsentiert stets ein Original, das jedoch auch wieder ein Modell sein kann), das Verkürzungsmerkmal (ein Modell erfasst nur relevante Eigenschaften des Originals) und das pragmatische Merkmal (ein Modell ist einem Original i. d. R. nicht eindeutig zugeordnet, kann aber als Ersatz für das Original für bestimmte Aufgaben verwendet werden). Die Ablauforganisation steht hierbei orthogonal zur Aufbauorganisation (Jablonski, 1994; Gaitanides, 2012). Eine Prozessinstanz ist eine Instanz eines Prozessmodells, d. i. eine konkrete Ausführung eines Prozesses. Ist der Prozess in einem Modell modelliert, sollte die Ausführung dem Modell nicht widersprechen. Prozessausführungen können in Ausführungsprotokollen (*Logs*) automatisiert aufgezeichnet werden; die einzelnen Instanzen sind somit rekonstruierbar (van der Aalst et al., 2004).

1.3.1 Routineprozesse und agile Prozesse

Es werden in der Realität nicht nur verschiedene Prozessmodellierungssprachen unterschieden, es existieren ebenso grundsätzlich zwei verschiedene Arten von Prozessen: Routineprozesse und agile Prozesse (Jablonski, 1994). Routineprozesse zeichnen sich dadurch aus, dass sie relativ vorhersehbar sind und somit alle Entscheidungsmöglichkeiten während der Ausführung zum Zeitpunkt der Modellierung bekannt sind und berücksichtigt werden können. Beispiele für Routineprozesse sind Fertigungsprozesse in der Industrie oder Abläufe in Banken und Versicherungen. Agile Prozesse hingegen erfordern eine Vielzahl an Entscheidungen, sodass zwar die Aufgaben bekannt sind, nicht aber ihre exakte Reihenfolge oder wer eine Aufgabe erledigen kann bzw. darf. Zu den agilen Prozessen zählen beispielsweise Design-Prozesse oder auch viele Prozesse im Gesundheitswesen (Reichert und Weber, 2012).

Entsprechend der beiden Arten von Prozessen, deren Übergang eher fließend zu sehen ist, gibt es auch die grundsätzliche Unterscheidung in imperative und deklarative Prozessmodellierung. Die imperative Modellierung ist hierbei für Routineprozesse geeignet und verfolgt den Ansatz, dass alles, was erlaubt ist, explizit im Modell erfasst sein muss. Der Kontrollfluss, der bei Routineprozessen vorab bekannt ist, legt die Reihenfolge der Aufgaben fest, wobei es durchaus alternative Reihenfolgen geben kann. Alle alternativen Reihenfolgen müs-

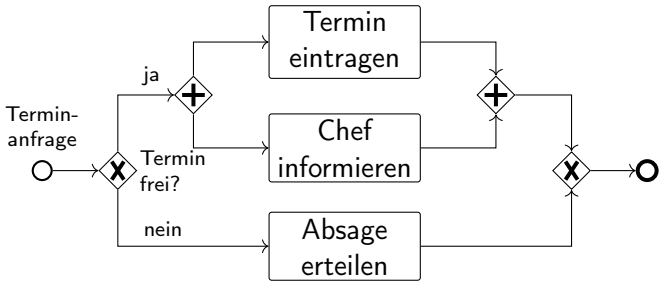


Abbildung 1.1: Beispiel für ein Prozessmodell in BPMN.

sen zum Zeitpunkt der Modellierung aber bekannt sein. Die deklarative Modellierung ist für agile Prozesse geeignet und definiert Regeln, die meist kausale Zusammenhänge der beteiligten Entitäten widerspiegeln und die bei der Prozessausführung nicht verletzt werden dürfen. Alles, was keiner Regel widerspricht, ist erlaubt. Durch die Einhaltung der Regeln ergibt sich indirekt ein Kontrollfluss, d. h. eine Menge an alternativen Reihenfolgen für die Aktivitäten, der allerdings mit Mitteln der imperativen Modellierung gar nicht oder zumindest nur schwer darstellbar wäre (Günther und van der Aalst, 2007). Fahland et al. (2009b) sehen den wesentlichen Unterschied zwischen imperativen und deklarativen Prozessmodellen darin, wie ermittelt werden kann, ob ein gegebenes Verhalten konform bezüglich des zugrunde liegenden Prozessmodells ist oder nicht. Im Fall von imperativen Modellen muss ein Verhalten mittels eines Pfades durch das Modell rekonstruiert werden können, im Fall von deklarativen Prozessmodellen wird geprüft, dass zu keinem Zeitpunkt eine der aufgestellten Prozessregeln verletzt ist.

1.3.2 Imperative Prozessmodellierung

Bekannte Vertreter imperativer Prozessmodellierungssprachen sind beispielsweise die Business Process Model and Notation (BPMN) (Object Management Group, 2011), die Eventgesteuerten Prozessketten (EPKs) (Scheer, 2002) und Petrinetze (van der Aalst und Stahl, 2011; Petri, 1962).

Abbildung 1.1 stellt ein Beispielmodell in BPMN dar. Es beschreibt den Prozess, wie eine Anfrage nach einem Termin mit dem Vorgesetzten von einer Bürokraft abgewickelt wird. Der Prozess beginnt mit einem Startereignis, beispielsweise einem eingehenden Anruf, der in BPMN durch einen Kreis mit dünner Umrandung gekennzeichnet wird. Die Bürokraft entscheidet sich dann an der folgenden Verzweigung (Raute mit $\times$-Symbol) für einen der nachfolgenden Wege: Entweder ist der angefragte Termin nicht möglich, dann wird eine Absage erteilt (Aktivität, ausgedrückt durch ein Rechteck), oder der Termin ist möglich. Im zweiten Fall können die anderen beiden Aktivitäten, „Termin eintragen“ und „Chef informieren“, unabhängig voneinander, d. h. insbesondere in einer beliebigen Reihenfolge oder zeitgleich, von der Bürokraft ausgeführt werden. Diese Unabhängigkeit ist durch die Raute mit $+$-Symbol gekennzeichnet. Alle Verzweigungen werden in umgekehrter Reihenfolge, in der sie begonnen werden, wieder vereinigt, ehe das Prozessmodell mit einem Endereignis (Kreis mit dicker Umrandung) das Prozessende markiert. Die Reihenfolge, in der die Modellelemente durchlaufen werden, wird dabei durch Sequenzflusspfeile vorgegeben.

Der grundsätzliche Ablauf für ein Prozessmodell als EPK ist derselbe wie für ein Modell in BPMN. Für EPKs gelten allerdings spezielle Regeln für die verwendeten Knoten. Eine EPK besteht mindestens aus drei verschiedenen Knotentypen: Ereignissen (Sechsecke), Funktionen

(Rechtecke mit abgerundeten Ecken) und Konnektoren (Kreise). Ereignisse und Funktionen müssen sich stets abwechseln, wobei ein Prozess immer mit einem Ereignis beginnt und endet. Entscheidungen, eine spezielle Art der Konnektoren, die durch ein XOR gekennzeichnet sind, können stets nur nach Funktionen geschehen, weswegen im Modell in Abbildung 1.2 auch die Funktion „Termin prüfen“ eingefügt ist, die im BPMN-Beispielmodell nicht vorhanden ist.

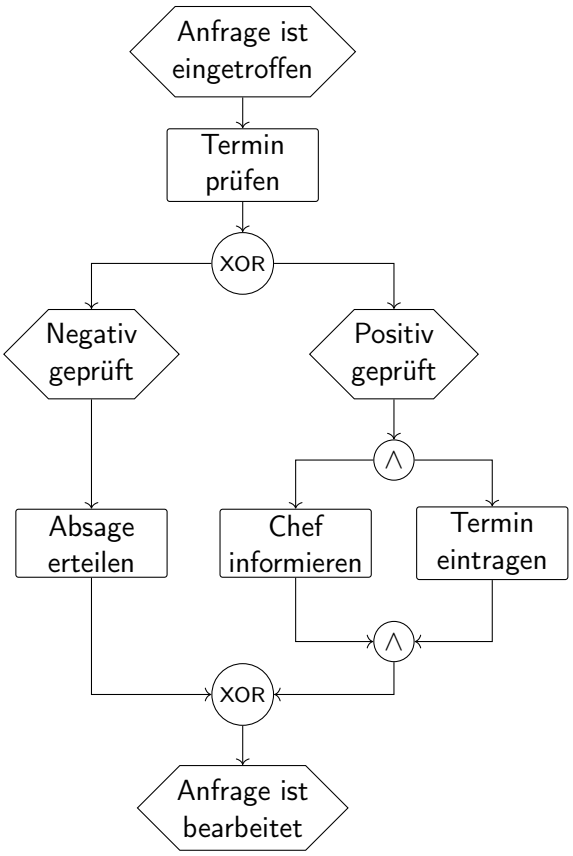


Abbildung 1.2: Beispiel für ein Prozessmodell als EPK.

Ein Petrinetz besteht aus zwei verschiedenen Arten von Knoten, aus Stellen (Kreise) und Transitionen (Rechtecke). Stellen und Transitionen sind über Kanten stets abwechselnd verbunden. Eine Transition ist dann aktiv, wenn alle eingehenden Stellen mit Markierungen (*tokens*) belegt sind. In Abbildung 1.3 ist die Stelle ganz links markiert, was hier bedeutet, dass eine Terminanfrage eingetroffen ist. Beide nachfolgenden Transitionen könnten schalten, d. h., beide Transitionen sind schaltfähig, da die Stelle jedoch nur genau eine Markierung hat, kann auch nur eine der beiden Transitionen tatsächlich schalten. Der Begriff „schalten“ bedeutet hierbei „ausgeführt/aktiviert werden“ und ist konkret das Entnehmen von Marken aus dem unmittelbaren Vorbereich der Transition und das Legen neuer Marken in den unmittelbaren Nachbereich. Schaltet die untere Transition „Absage erteilen“, ist der Prozess sofort beendet. Schaltet die obere Transition, werden die beiden nachfolgenden Stellen markiert und sowohl „Termin eintragen“ und „Chef informieren“ sind schaltfähig und können auch beide gleichzeitig schalten. Erst wenn diese beiden Transitionen geschaltet haben, ist die letzte unbeschriftete Transition aktiviert, die nach ihrem Schalten die letzte Stelle markiert.

Um Methoden zum Ähnlichkeitsabgleich von Prozessmodellen auch auf Modelle unterschiedlicher Sprachen anwenden zu können, ist es notwendig, Prozessmodelle in abstrakterer

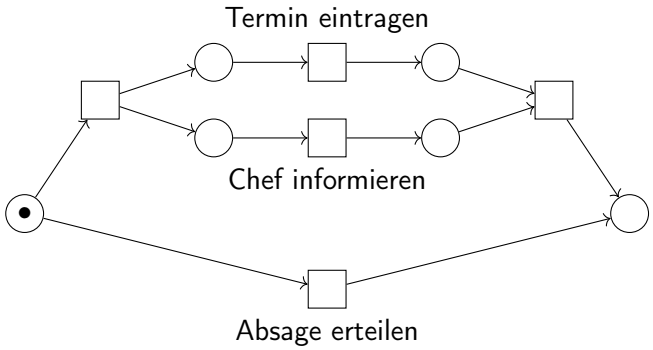


Abbildung 1.3: Beispiel für ein Prozessmodell als Petrinetz.

Form als den Modellierungssprachen an sich zu definieren. Zwar gibt es Ähnlichkeitsmaße, die beispielsweise direkt auf EPKs arbeiten (z. B. Dijkman et al., 2011), jedoch ist deren Anwendbarkeit dann auf derartige Modelle beschränkt. Von Dijkman et al. (2009b) wird eine allgemeinere, abstraktere Definition von Prozessmodell, der Prozessgraph, wie folgt vorgeschlagen:

Definition 1.1 (Imperatives Prozessmodell I)**.** Sei $\mathcal{L} \subseteq \{s_1 s_2 \dots s_n \mid s_i \text{ ist ein Zeichen } \forall i \in \{1, 2, \dots, n\}, n \in \mathbb{N}\}$ eine Menge an Beschriftungen. Ein imperatives Prozessmodell G ist ein Tupel $G = (N, E, \lambda)$, wobei

- N eine Menge an Knoten ist,
- $E \subseteq N \times N$ eine Menge an Kanten ist und
- $\lambda : N \to \mathcal{L}$ eine Funktion ist, die Knoten auf Beschriftungen abbildet.

Auf diese Definition lassen sich unter anderem auch BPMN-Modelle, EPKs und Petrinetze zurückführen, wobei auffällt, dass die abstrakte Definition nur eine Art Knoten zulässt, während sich beispielsweise in BPMN viele unterschiedliche Arten an Knoten modellieren lassen, unter anderem Aktivitäten, Gateways und Events. Diese Typen können über die Labelfunktion λ abgefangen werden, indem bei Knoten ohne Beschriftung, also typischerweise bei Gateways und Events, der entsprechende Typ als Beschriftung ausgegeben wird. Speziell für Gateways, also für Kontrollflussverzweigungen, besteht auch die Möglichkeit, diese aus dem abstrakten Prozessmodell auszublenden und stattdessen die damit verbundenen Informationen an die noch verbliebenen Modellelemente anzuheften (vgl. Minor et al., 2007). Ebenso wäre es möglich, die Definition des imperativen Prozessmodells zu verfeinern, um weitere Knotentypen explizit zu berücksichtigen (Dijkman et al., 2009a). Um allerdings eine Allgemeingültigkeit der darauf aufbauenden Abgleichsmethoden zu erhalten, sollte diese Verfeinerung nicht zu stark sein. Um BPMN-Modelle, EPKs und Petrinetze weiterhin zu berücksichtigen, ist die folgende Verfeinerung der Knotenmenge angemessen:

$$N = A \dot\cup C \dot\cup \{e_{start}\} \dot\cup \{e_{end}\} \text{ mit } C = XOR_s \dot\cup XOR_j \dot\cup AND_s \dot\cup AND_j. \tag{1.1}$$

Die Menge A bezeichnet hierbei Aktivitäten, XOR_s sind exklusive Verzweigungen, XOR_j die Zusammenführungen von exklusiv ausgeführten Zweigen, AND_s sind parallele Verzweigungen und AND_j deren Zusammenführungen. Es werden sowohl die Begriffe Verzweigungsknoten als auch Gateways für die Elemente aus C verwendet. Der Knoten e_{start} bezeichnet

das Startereignis und e_{end} das Endereignis. Der oftmals ebenfalls angeführte dritte Verzweigungstyp, das OR-Gateway oder inklusive Gateway, kann aus XOR- und AND-Gateways nachgebildet werden, weshalb es hier nicht separat aufgeführt ist. Es wird auch in der Praxis, wie zur Muehlen und Recker (2013) zeigen, nicht häufig verwendet.

Um die Ähnlichkeit von Prozessen in der Business Process Execution Language (BPEL) (OASIS, 2007), einer textuellen, blockstrukturierten Prozessausführungssprache basierend auf der eXtensible Markup Language (XML), die vor allem zur Orchestrierung von Webservices dient, zu bestimmen, wandeln Grigori et al. (2010) die in BPEL spezifizierten Prozesse ebenfalls in Prozessgraphen (*Behavioral Graphs*) gemäß Definition 1.1 mit der Knotenverfeinerung in Gleichung (1.1) um. Dies unterstreicht die Verwendung der abstrakten Darstellung imperativer Prozessmodelle. Unter anderem entsprechen auch die von Bae et al. (2006b) genannten Abhängigkeitsgraphen (*Dependency Graphs*) oder die von Weidlich et al. (2010b) verwendeten Workflow-Netze den Modellen aus Definition 1.1.

Zur Darstellung der Prozessgraphen aus Definition 1.1 wird in dieser Arbeit eine BPMN-ähnliche Notation wie in Abbildung 1.1 verwendet. Das heißt, dass insbesondere die verschiedenen Typen an Knoten aus Gleichung (1.1) mit den grafischen Mitteln der BPMN unterschieden werden.

1.3.3 Deklarative Prozessmodellierung

Zu den deklarativen Prozessmodellierungssprachen zählen beispielsweise DECLARE (Pesic et al., 2007) bzw. dessen grafische Repräsentation ConDec (Pesic und van der Aalst, 2006), DCR Graphen (Mukkamala, 2012), die Case Management Model and Notation (CMMN) (Object Management Group, 2014) oder die Declarative Process Intermediate Language (DPIL) (Zeising et al., 2014). Ihnen gemein ist die folgende Definition eines deklarativen Prozessmodells.

Definition 1.2 (Deklaratives Prozessmodell I)**.** Ein deklaratives Prozessmodell $S = (A, C)$ besteht aus einer endlichen Menge an Aktivitäten A und einer endlichen Menge an Regeln C, welche die Randbedingungen der Prozessausführung vorgeben.

Die Aktivitäten werden hierbei direkt mit ihrer Beschreibung identifiziert. Oft wird bei den Regeln der Menge C unterschieden, ob eine Regel verpflichtend oder optional ist, d. h., ob eine Regel bei der Ausführung unbedingt eingehalten werden muss oder ob sie im Sinne einer Empfehlung einhalten werden sollte, aber auch (bewusst) von ihr abgewichen werden kann. Die Menge der verpflichtenden Regeln wird mit C_M bezeichnet und die der optionalen Regeln mit C_O. Es gilt dann $C = C_M \dot{\cup} C_O$. Wie die Regeln aus C konkret formuliert sind, hängt von der gewählten Modellierungssprache ab. In DECLARE basieren die Regeln auf linearer temporärer Logik (LTL) (Pesic, 2008), in DPIL hingegen auf Prädikatenlogik erster Ordnung (FOL) (Zeising et al., 2014). LTL ist äquivalent zur einstelligen FOL (*monadic first order logic*), d. h., in FOL lassen sich alle Aussagen der LTL ausdrücken, aber nicht umgekehrt (Kamps Theorem, Kamp, 1968). Oft gibt es für die Formulierung von Prozessen vorgefertigte Makros, um nicht direkt mit Logikausdrücken modellieren zu müssen. Diese Makros basieren in der Regel auf Workflow Patterns (van der Aalst et al., 2003a), also bestimmten Konstrukten bzw. Mustern, die in Prozessen immer wieder auftauchen und aus denen sich Prozessmodelle zusammensetzen lassen.

In Abbildung 1.4 ist der Prozess der Terminanfrage in ConDec modelliert, einer Menge an vorgefertigten Makros für DECLARE. In diesem Modell werden genau drei verschiedene

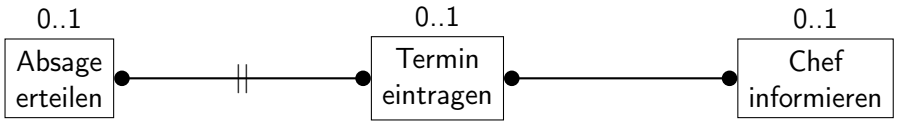


Abbildung 1.4: Beispiel für ein Prozessmodell in DECLARE/ConDec.

Regeln verwendet. Die Regel zwischen „Absage erteilen“ und „Termin eintragen“ besagt, dass nicht beide Aktivitäten in einer Prozessinstanz, also während einer Ausführung, ausgeführt werden dürfen (not coexistence). Die Regel zwischen „Termin eintragen“ und „Chef informieren“ besagt hingegen, dass wenn eine der beiden Aktivitäten ausgeführt wird, auch die andere in derselben Instanz ausgeführt werden muss, ohne dabei eine Reihenfolge zu spezifizieren (coexistence). Die Angabe 0..1 über den Aktivitäten meint, dass die jeweiligen Aufgaben maximal einmal ausgeführt werden können (existence(0..1)). Ein Weglassen der 0..1-Regel für die Aktivität „Chef informieren“ hätte zur Folge, dass diese Aktivität beliebig oft ausgeführt werden könnte – wobei sie, wenn „Termin eintragen“ stattgefunden hat, auf jeden Fall mindestens einmal ausgeführt sein/werden muss. Dies macht deutlich, dass deklarativ modellierte Prozesse variantenreicher werden, je weniger Regeln im Modell spezifiziert sind. Im Modell in Abbildung 1.4 ist beispielsweise nicht bestimmt, dass überhaupt eine der Aktivitäten ausgeführt werden muss. Möchte man diese Einschränkung hinzufügen, so müsste dies durch Hinzufügen einer weiteren Regel, in diesem Fall einer 1 of 2-Regel zwischen „Absage erteilen“ und „Termin eintragen“ oder „Absage erteilen“ und „Chef informieren“, geschehen. Das bedeutet, dass mindestens eine der Aktivitäten ausgeführt werden muss. Abhängig von dieser Ausführung müssen oder dürfen dann weitere Aktivitäten ausgeführt werden oder nicht. Bei imperativ modellierten Prozessen verhält es sich genau umgekehrt. Um beispielsweise eine beliebigfache Wiederholung von „Chef informieren“ im Modell aus Abbildung 1.1 zu erlauben, müssten weitere Modellkonstrukte, z. B. eine Schleife, hinzugefügt werden, was das Modell umfangreicher machen würde.

1.4 Ziel dieser Arbeit

Das Ziel dieser Arbeit ist es, für die verschiedenen Arten an Prozessmodellen eine Auswahl an Abgleichsmethoden zu entwickeln, um damit die in Abschnitt 1.1 aufgeführten Motive für einen Ähnlichkeitsabgleich von Prozessmodellen erfüllen zu können. Der Herausforderung, die sich durch eine uneinheitliche Terminologie ergibt, soll durch das Einbeziehen von weiteren Modellinformationen, die über die Beschreibung der Aufgaben hinausgehen, begegnet werden. So sollen Informationen über zuständige Agenten (Personen, Rollen), verwendete Datenobjekte, verwendete Werkzeuge/Services und bestimmte Verhaltensregeln wie Exklusivität oder Unabhängigkeit von Aufgaben berücksichtigt werden. Das Problem der unterschiedlichen, weil nicht festgelegten, Granularität soll durch eine spezielle Abbildungsvorschrift, eine Zuordnung der Aktivitäten auf Mengenbasis, gelöst werden. Abbildung 1.5, die die Notation der BPMN verwendet, zeigt zwei Prozessmodelle, für die ein sogenannter 1:1-Abgleich angebracht ist, also eine Abbildung, die Korrespondenzen zwischen einzelnen Aktivitäten festlegt. Es kann hier für jede Aktivität aus dem linken Modell eine entsprechende Aktivität im rechten Modell gefunden werden. Da im Allgemeinen aber nicht vorher bekannt ist, dass tatsächlich, sofern Entsprechungen in den abzugleichenden Modellen enthalten sind, diese eindeutig sind, insbesondere wenn die Prozessmodelle in unterschiedlicher Granularität modelliert sind, sollten Korrespon-

denzen in den Modellen so gesucht werden, dass auch mehrere Aktivitäten in einem Modell mehreren Aktivitäten im anderen Modell entsprechen können. In Abbildung 1.6 ist solch ein Fall einer M:N-Abbildung gezeigt. Es ist hier auch nicht möglich bzw. nicht sinnvoll, die Korrespondenzen in 1:N-Entsprechungen aufzulösen, also dass eine Aktivität im einen Modell mehreren Aktivitäten im anderen Modell entspricht. Ein Beispiel für solch einen Sachverhalt ist in Abbildung 1.7 gezeigt. 1:N-Entsprechungen können in nur eine Richtung existieren, dann ist ein Modell eine echte Verfeinerung des anderen Modells, oder sie können in beide Richtungen existieren, wie in Beispielabbildung 1.7 gezeigt. Um die Beispielmodelle übersichtlicher zu gestalten, ist für die einzelnen Aktivitäten nur die Aufgabenbeschreibung angegeben ohne zuständige Agenten, Datenobjekte oder Werkzeuge zu nennen. 1:1- und 1:N-Abbildungen sind, wie in Kapitel 3 gezeigt wird, Spezialfälle der allgemeineren M:N-Abbildung.

Wenn möglich sollen für die zu entwickelnden Abgleichsmethoden, die M:N-Korrespondenzen erlauben und die die unterschiedlichen Modellinformationen berücksichtigen, bestehende Methoden aus der Literatur, die ausreichend evaluiert sind, übernommen und angepasst werden. Wenn eine solche Anpassung nicht möglich ist, müssen eigene Abgleichsverfahren gefunden werden, deren Funktionsweise mittels einer prototypischen Implementierung und eines Vergleichs mit bestehenden Methoden demonstriert wird. Bei den zu entwickelnden Abgleichmethoden stellt sich insbesondere die Frage, ob für jede Abbildungsart (1:1, 1:N, M:N), für jede Modellinformation (Beschreibung, Agenten, Datenobjekte, Werkzeuge, Verhalten) und für die beiden unterschiedlichen Modellierungsarten (imperativ, deklarativ) eine eigene Methode zu finden ist oder ob Methoden zwischen verschiedenen Anwendungsfällen übertragen werden können. Eine weitere interessante Fragestellung, die sich aus der oben genannten Frage ergibt, ist die, ob allgemein ein Vorgehen identifiziert werden kann, wie ein Ähnlichkeitsabgleich durchzuführen ist. Genauer: Welche einzelnen Schritte müssen bei einem Abgleich erledigt werden, wie hängen diese Schritte zusammen und wo bzw. wie können die Schritte unabhängig voneinander an die jeweiligen Gegebenheiten angepasst werden? Auf diese letzte Frage wird während der Arbeit bei der Untersuchung der anderen Ziele immer wieder eingegangen, sodass sich auch hier zum Schluss ein Gesamtbild über das allgemeine Vorgehen ergibt.

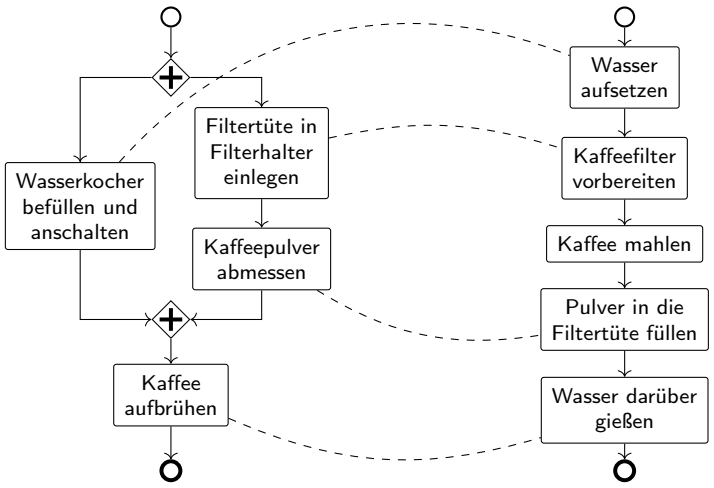


Abbildung 1.5: Beispiel für zwei Prozessmodelle mit 1:1-Entsprechungen.

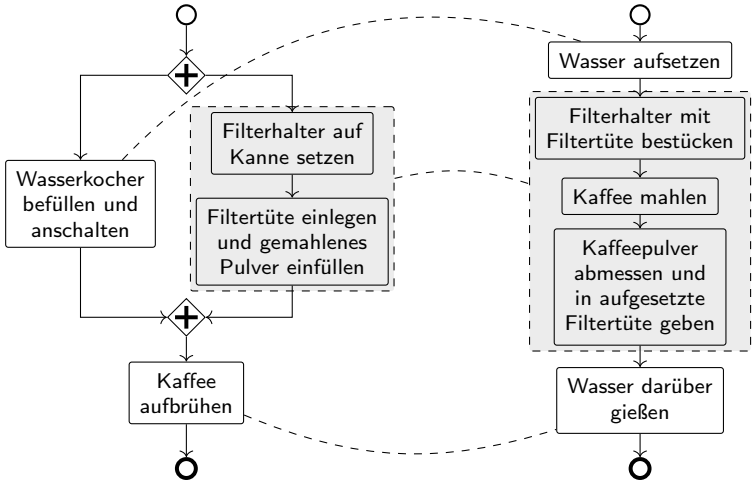


Abbildung 1.6: Beispiel für zwei Prozessmodelle mit M:N-Entsprechungen, genauer: einer 2:3-Entsprechung.

1.5 Forschungsmethode

Die Forschungsmethodik dieser Arbeit entspricht der konstruktiven Forschung (*constructive research*), die Probleme durch die Konstruktion von Modellen, Methoden, Diagrammen, Plänen usw. löst (Kasanen et al., 1993). Die konstruktive Forschung ist in der (Wirtschafts-) Informatik weit verbreitet (Frank, 2006). Die Forschungsmethodik lässt sich grob in sechs Phasen unterteilen (Kasanen et al., 1993; Oyegoke, 2011), deren Reihenfolge nicht unbedingt auf die folgende festgelegt ist:

1. Finde ein relevantes Praxisproblem, das zudem Forschungspotential beinhaltet.
2. Gewinne ein generelles und umfassendes Verständnis der Thematik.
3. Führe eine Lösungsidee ein, d. h., konstruiere diese.
4. Zeige, dass die Lösung funktioniert.
5. Lege die theoretischen Zusammenhänge und den Forschungsbeitrag des Lösungskonzepts offen.
6. Untersuche den Anwendungsbereich der Lösung.

Diese Phasen finden sich in der vorliegenden Arbeit, also im gezeigten Lösungsansatz, wie folgt wieder. Die Relevanz des Praxisproblems, Prozessmodelle untereinander abzugleichen und deren Ähnlichkeit zu bestimmen, wird zum einen in vielen verwandten Arbeiten thematisiert, zum anderen auch durch die Forschung am Lehrstuhl für Angewandte Informatik IV der Universität Bayreuth als notwendig empfunden. Besonders durch das Aufkommen multiperspektivischer, regelbasierter Modelle ist ein Abgleich von Prozessmodellen zum besseren Verständnis dieser Modelle hilfreich (Phase 1). Eine umfangreiche Analyse verwandter Arbeiten bietet einen Einblick in das grundsätzliche Vorgehen beim Prozessmodellabgleich und zeigt die oft sehr unterschiedlichen Ansätze, mit denen bisher ein Ähnlichkeitsabgleich durchgeführt wird (Phase 2). Eine Lösungsidee, d. h. Abgleichsmethoden für multiperspektivische Prozessmodelle sowie deklarative Prozessmodelle, wird analog zu bisherigen Lösungen definiert, wobei

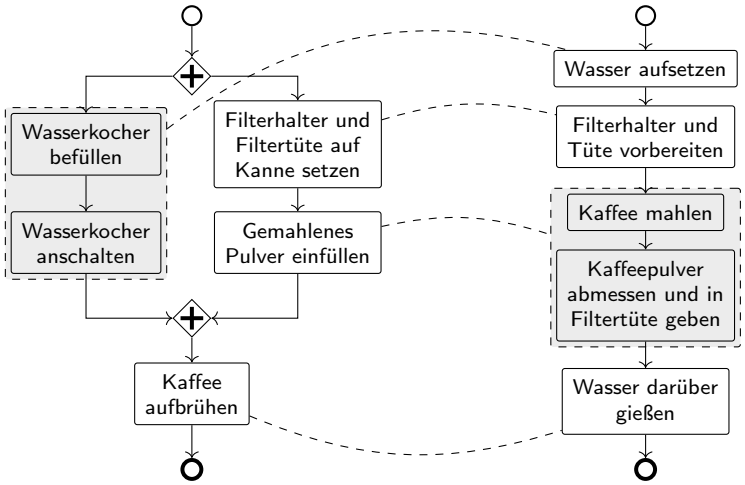


Abbildung 1.7: Beispiel für zwei Prozessmodelle mit 1:N-Entsprechungen ohne echte M:N-Entsprechungen.

teilweise bestehende Methoden erweitert werden, teilweise auch neue Ansätze, die den grundsätzlich geforderten Bedingungen für Ähnlichkeitsmaße genügen, entwickelt werden (Phase 3). Das Funktionieren der Lösung wird mittels einer prototypischen Implementierung gezeigt. Vergleiche mit bisherigen Abgleichsmethoden werden ebenfalls durchgeführt (Phase 4). Auf die Verknüpfung zu verwandten Arbeiten und bestehenden theoretischen Konzepten und die analogen Erweiterungen wird an den jeweiligen Stellen hingewiesen. Der Forschungsbeitrag ergibt sich aus der Aufspannung aller möglichen Abgleichsfelder, unter anderem der Unterscheidung imperativer und deklarativer Modelle, und den neu definierten Abgleichsmethoden (Phase 5). Der Anwendungsbereich ergibt sich direkt aus den Definitionen. Eine Anwendung kann für alle Prozessmodelle und Ressourcen erfolgen, die den geforderten, grundsätzlich sehr allgemein gehaltenen Definitionen genügen. Auf Einschränkungen der Anwendbarkeit wird an den jeweiligen Stellen hingewiesen (Phase 6). Die konstruktive Forschungsmethodik ist verwandt mit der Design Science (Hevner et al., 2010) bzw. ein Teil der Design Science (Frank, 2006). Die Design Science erweitert die rein konstruktive Forschungsmethodik um die Forderung nach der Präsentation der Ergebnisse, beispielsweise durch Veröffentlichung in wissenschaftlichen Werken, wobei auch nicht wissenschaftliches Publikum explizit angesprochen werden soll, um den anwendungsrelevanten Charakter hervorzuheben. Teilergebnisse der vorliegenden Arbeit wurden auf verschiedenen Konferenzen und Kolloquien, unter anderem dem BPM-Workshop EOMAS 2014, dem ICSOC-Workshop RMSOC 2015 und dem Informatik-Workshop ZuGPM 2016, bereits vorgestellt und veröffentlicht. Die genauen Daten können dem Literaturverzeichnis entnommen werden.

1.6 Aufbau der Arbeit

Kapitel 2 stellt verschiedene bisherige Abgleichsmethoden aus der Literatur für imperative Prozessmodelle vor, die vornehmlich auf den Aktivitätenbeschreibungen, der Ablauflogik/-dem Prozessverhalten und der Struktur der (imperativen) Prozessmodelle beruhen. Zentrales Element ist hierbei der vierstufige Ansatz, ein Optimierungsverfahren, das auf Basis von Abbildungen zwischen den zu vergleichenden Prozessmodellen diejenigen Korrespondenzen

zwischen den Modellen aufdeckt, die den größten Ähnlichkeitswert liefern.

In Kapitel 3 werden zunächst die Grenzen der aus der Literatur entnommenen, bisherigen Methoden aufgezeigt und notwendige Erweiterungen vorgestellt. Diese Erweiterungen betreffen die verschiedenen Perspektiven, die ein Prozess beinhaltet: Neben der Beschreibung der Aktivitäten und der Ablauflogik sind das die beteiligten Personen/Rollen/Gruppen, die verwendeten Dokumente und Datenobjekte sowie die benutzten Werkzeuge und Services (Abschnitt 3.2). Außerdem werden die Methoden hinsichtlich der erlaubten Abbildung, also der erlaubten Korrespondenzenbildung erweitert, sodass auch unterschiedlich granulare Prozessmodelle miteinander abgeglichen werden können, ohne dass aufgrund der eingeschränkten Definition von vornherein niedrige Ähnlichkeitswerte erwartet werden müssen. Statt ausschließlich Abbildungen zuzulassen, die jeweils eine Aktivität einer anderen zuordnen (1:1-Abbildungen), werden solche Abbildungen eingeführt, die Mengen von Aktivitäten auf Mengen von Aktivitäten abbilden. Diese Abbildungen werden M:N-Abbildungen genannt (Abschnitt 3.3).

Kapitel 4 stellt neue Abgleichsmethoden für die erweiterten Definitionen des vorhergehenden Kapitels vor. Je nach den äußerlichen Begebenheiten kann hierbei teilweise zwischen alternativen Methoden gewählt werden, um für die jeweilige Situation ein möglichst genaues Ähnlichkeitsmaß zu erhalten. Diese Erweiterungen beziehen sich sowohl auf imperative als auch auf deklarative Prozessmodelle, wobei gezeigt wird, wie ein Teil der Methoden von imperativen auf deklarative Prozessmodelle übertragen werden kann. Vor allem was die Ablauflogik betrifft, müssen für deklarative Modelle neue Methoden gefunden werden. Für die anderen Perspektiven können Analogien zwischen den zwei grundsätzlichen Modellarten für eine Übertragbarkeit genutzt werden.

Kapitel 5 stellt anschließend eine prototypische Proof of Concept-Implementierung sowie Evaluationen der einzelnen neuen Ansätze vor. Insbesondere wird ein Vergleich mit Modellen, die im Rahmen des Process Model Matching Contests zu analysieren sind, gezogen. Die Implementierung setzt sich aus zwei grundlegend verschiedenen Bausteinen zusammen, nämlich einem Python-Programm, das die Eingabemodelle vorverarbeitet, und einem ZIMPL-Programm, das die endgültige Berechnung des Ähnlichkeitswerts, der die Lösung eines Optimierungsproblems darstellt, innerhalb der SCIP Optimization Suite übernimmt.

Die Arbeit schließt mit Kapitel 6, das die Arbeit kurz zusammenfasst, Grenzen des neu entwickelten Lösungsansatzes offenlegt und Richtungen für zukünftige Forschungsarbeiten aufzeigt.

Kapitel 2

Ähnlichkeitsabgleich in verwandten Arbeiten

In der Literatur werden einige Verfahren für einen Ähnlichkeitsabgleich von Prozessmodellen genannt, wobei solch ein Abgleich bislang vornehmlich auf imperativen Prozessmodellen durchgeführt wird. Dies mag an der Verbreitung der imperativen Modelle vor allem in der Praxis liegen. Erste Ansätze für deklarative Modelle sind von Baumann et al. (2016a) genannt oder beruhen auf einer Überführung der deklarativen Modelle in imperative Modelle oder Zustandsautomaten, doch Näheres hierzu findet sich in Kapitel 4.5. Die verschiedenen Abgleichsmethoden für imperative Modelle, die in der Literatur zu finden sind, lassen sich hierbei grob in drei Bereiche aufteilen: Methoden, die die Beschriftung von Prozessmodellen ausnutzen, Methoden, die das Verhalten von Prozessmodellen verwenden, und Methoden, die auf der Struktur von Prozessmodellen aufbauen. Die folgenden Abschnitte stellen einen umfassenden, wenn auch nicht erschöpfenden Überblick über die gängigsten Abgleichsmethoden vor, die in verwandten Arbeiten genannt werden. Hierbei bieten beispielsweise auch die Arbeiten von Schoknecht et al. (2017), Becker und Laue (2012), Dijkman et al. (2011) und Wombacher und Rozie (2006) jeweils einen vergleichenden Überblick über ausgewählte Verfahren.

Abschnitt 2.1 stellt den Begriff der Ähnlichkeit zunächst allgemein vor, ehe eine Definition für Ähnlichkeitsmaße, die auch für Prozessmodelle gilt, gegeben wird. Außerdem wird kurz darauf eingegangen, wann zwei Prozessmodelle unter Verwendung eines Ähnlichkeitsmaßes gleich sind. Abschnitt 2.2 stellt anschließend Abgleichsverfahren für Prozessmodelle vor, die in der Literatur genannt sind. Hierfür wird zunächst das grunsätzliche, vierstufige Verfahren erläutert, nach dem bei einer Ähnlichkeitsbestimmung vorgegangen wird (Abschnitt 2.2.1). Anschließend werden labelbasierte Ansätze (Abschnitt 2.2.2), strukturbasierte Ansätze (Abschnitt 2.2.3) und verhaltensbasierte Ansätze (Abschnitt 2.2.4), die auf dem vierstufigen Verfahren aufbauen, vorgestellt. Um zu zeigen, dass es über das vierstufige Verfahren hinaus auch andere Möglichkeiten gibt, Ähnlichkeiten zwischen Prozessmodellen zu bestimmen, werden in Abschnitt 2.3 einige davon angesprochen. Abschnitt 2.4 schließt dieses Kapitel mit einer kurzen Zusammenfassung und einer Einordnung der der Literatur entnommenen Abgleichsmöglichkeiten.

2.1 Ähnlichkeiten

Der Begriff des Ähnlichkeitsmaßes findet hauptsächlich in der Statistik und verwandten Feldern Verwendung. Ein Ähnlichkeitsmaß beschreibt eine reellwertige Funktion, die die Ähnlichkeit zweier Objekte quantifiziert. Hierbei gibt es keine allgemeingültige Definition eines Ähnlichkeitsmaßes. In den meisten Fällen wird jedoch eine Art Inverse einer Distanzmetrik zur Messung von Ähnlichkeiten herangezogen. Große Werte deuten dabei große Ähnlichkeit der zu vergleichenden Objekte an, während Werte gegen null oder gar negative Werte Unähnlichkeit ausdrücken. Ein paar prominente Beispiele für Ähnlichkeitsmaße aus dem statistischen Umfeld werden im Folgenden genannt. Im Bereich der Clusteranalyse wird von Frey und Dueck (2007) folgendes Ähnlichkeitsmaß aufbauend auf dem quadrierten Euklid'schen Abstand vorgeschlagen (Huang, 2008):

$$sim_{eukl}(x, y) = -\|x - y\|_2^2 \in (-\infty, 0]$$

Der Ausdruck $\|\cdot\|_2$ bezeichnet die 2-Norm (euklidische Norm) und ist definiert als $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$, $x \in \mathbb{C}^n$.[1] Für reellwertige Vektoren wird häufig die Kosinus-Ähnlichkeit (Huang, 2008) als Maß für deren Ähnlichkeit herangezogen. Es wird der Kosinus des Winkels zwischen den beiden zu vergleichenden Vektoren bestimmt. Weisen die Vektoren in etwa in dieselbe Richtung, wobei hier sprachlich nicht zwischen den Begriffen Richtung und Orientierung unterschieden wird, so ist der Zwischenwinkel nahe null und der Kosinus des Winkels nahe bei eins. Stehen die Vektoren im rechten Winkel zueinander, d. h., sind die Vektoren unabhängig voneinander, so ist der Kosinus null. Deuten die Vektoren in entgegengesetzte Richtungen, geht der Kosinus gegen minus eins. Für $x, y \in \mathbb{R}^n$ ist

$$sim_{cos}(x, y) = \frac{\langle x, y \rangle}{\|x\|_2 \|y\|_2} = \cos\theta \in [-1, 1],$$

wobei θ den Zwischenwinkel zwischen x und y bezeichnet. Diese Art der Ähnlichkeitsbestimmung wird oft im Bereich des Textminings, beim Vergleich von Dokumenten, Multimediaobjekten oder auch in der Kryptographie angewendet. Der Ausdruck $\langle \cdot, \cdot \rangle$ bezeichnet das Standardskalarprodukt: $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$, $x, y \in \mathbb{R}^n$.

Im Gebiet des maschinellen Lernens, zum Beispiel bei Support Vector Machines oder neuronalen Netzen, werden sogenannte Kern-Funktionen zur Bestimmung der Ähnlichkeit verwendet (Schölkopf, 2001). Kern-Funktionen gibt es verschiedene, zum Beispiel lineare Kerne $k(x, y) = \langle x, y \rangle \in \mathbb{R}$, polynomiale Kerne $k(x, y) = \langle x, y \rangle^d \in \mathbb{R}$ für d ungerade bzw. $\in \mathbb{R}_0^+$ für d gerade oder auch den sogenannten (Gauß'schen) RBF-Kern $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \in \mathbb{R}^+$, wobei $\sigma > 0$ ein freier Parameter ist. Die Abkürzung RBF steht hierbei für radiale Basisfunktion und bezeichnet eine reellwertige Funktion, deren Wert nur vom Abstand zum Ursprung abhängt und somit radialsymmetrisch ist. Entsprechend unterschiedlich sind hier auch die Wertebereiche der jeweiligen Ähnlichkeitsmaße. Für die Ähnlichkeitsmessung von Prozessmodellen, die a priori nicht als Vektoren im $\mathbb{R}^n$ vorliegen, werden Ähnlichkeitsmaße benötigt, die Prozessmodelle oder zumindest bestimmte Aspekte von Prozessmodellen in Beziehung setzen und vergleichen können. Insofern sind die drei genannten Maße nicht für einen Ähnlichkeitsabgleich von Prozessmodellen geeignet. Außerdem verletzen sie, wie in Abschnitt 2.1.1 ersichtlich, die Definition eines Ähnlichkeitsmaßes (Definition 2.1). Lediglich die

[1] Für reellwertige Vektoren ist $|x_i|^2 = x_i^2$. Die Betragsstriche in der Definition der 2-Norm werden dann benötigt, wenn x echt komplex ist.

Kosinus-Ähnlichkeit kann unter bestimmten Voraussetzungen für einen Ähnlichkeitsabgleich von Prozessmodellen herangezogen werden, wozu sich auch in der Literatur Beispiele finden; in Abschnitt 2.2.4.4 ist eines aufgeführt. Ausgehend von den oben genannten, nicht geeigneten Ähnlichkeitsmaßen, werden im folgenden Abschnitt Ähnlichkeitsmaße definiert, wie sie im Fortlauf der Arbeit zu verstehen sind.

2.1.1 Ähnlichkeitsmaße

Im Allgemeinen wird eine Funktion als Ähnlichkeitsmaß bezeichnet, wenn sie den folgenden Eigenschaften[2] genügt (vgl. Richter, 1993):

Definition 2.1 (Ähnlichkeitsmaß)**.** Eine Funktion $sim : I \times I \to \mathbb{R}$, wobei I eine beliebige Eingabemenge ist, wird als Ähnlichkeitsmaß bezeichnet, wenn sie folgende Eigenschaften für alle $A, B \in I$ erfüllt:

- Nicht-Negativität: $sim(A, B) \geq 0$
- Symmetrie: $sim(A, B) = sim(B, A)$
- Identität: $A = B \Rightarrow sim(A, B) = 1$

Diese Definition von Ähnlichkeitsmaß ist das Gegenstück zur Definition einer Distanz bzw. eines Distanzmaßes und wird deswegen oft, wenn auch nicht immer, wie obige Gegenbeispiele zeigen, herangezogen. Die uneinheitliche Auffassung der Definition von Ähnlichkeit liegt unter anderem daran, dass in den meisten Fällen keine absoluten Ähnlichkeitswerte benötigt werden, sondern lediglich Ordnungen. In Abschnitt A.1 ist die Definition eines Distanzmaßes aufgeführt.

Für die Vergleichbarkeit verschiedener Ähnlichkeitsmaße für Prozessmodelle und wegen der Notwendigkeit, verschiedene Maße zu kombinieren, sollen Ähnlichkeitsmaße für Prozessmodelle normiert sein, d. h. stets Werte aus dem Intervall $[0, 1]$ annehmen:

$$sim : P \times P \to [0, 1], \tag{2.1}$$

wobei P eine Menge an Prozessmodellen ist. Dies ist im Allgemeinen durch eine Reskalierung erreichbar. Ein Wert von null bedeutet dann maximale Unähnlichkeit, während ein Wert von eins auf größtmögliche Ähnlichkeit hinweist. Haben zwei Objekte Ähnlichkeit eins, müssen diese jedoch nicht zwangsläufig gleich sein. Dies wird dadurch deutlich, dass die Identitätseigenschaft in dieser Definition nur in eine Richtung gegeben ist. Die Forderung $sim(A, B) = 1 \Rightarrow A = B$ ist oftmals zu restriktiv und trifft für viele existierende Ähnlichkeitsmaße in der Literatur nicht zu. Dies liegt daran, dass viele Ähnlichkeitsmaße die zu vergleichenden Prozessmodelle auf bestimmte Charakteristika reduzieren und somit Unterschiede außerhalb dieser Charakteristika nicht berücksichtigen. In den in Abschnitt 2.2 beschriebenen Methoden wird dies ebenfalls der Fall sein.

Zusätzlich ist eine weitere Eigenschaft für (normierte) Ähnlichkeitsmaße hilfreich, gerade wenn es darum geht, viele Modelle in großen Repositorien paarweise miteinander zu vergleichen:

- *Dreiecksungleichung:* $sim(A, C) \geq sim(A, B) + sim(B, C) - 1$

[2] Es wird hier bereichts ersichtlich, dass die genannten Beispiele – quadrierter euklidischer Abstand, Kosinus-Ähnlichkeit, Kern-Funktionen – keine Ähnlichkeitsmaße im Sinne dieser Definition sind.

Für normierte Ähnlichkeitsmaße, also Metriken, kann aus der Ähnlichkeitsfunktion *sim* eine Distanzfunktion mittels $d = 1 - sim$ gebildet werden.[3] Aus der Dreiecksungleichung für diese Distanzfunktion,

$$\begin{aligned} & d(A,C) \leq d(A,B) + d(B,C) \\ \Leftrightarrow\ & 1 - sim(A,C) \leq (1 - sim(A,B)) + (1 - sim(B,C)) \end{aligned}$$

ergibt sich die obige Bedingung für die Dreiecksungleichung der normierten Ähnlichkeitsfunktion. Erfüllt ein Ähnlichkeitsmaß die Identitätseigenschaft in beide Richtungen und die Dreiecksungleichung, kann man auch von einer Ähnlichkeitsmetrik sprechen.

Um aus einer beliebigen Distanzfunktion d ein Ähnlichkeitsmaß *sim* zu bilden, kann die Transformation

$$sim = \frac{1}{1+d} \tag{2.2}$$

angewendet werden (Becker und Laue, 2012). Oft lassen sich, wenn der Maximalwert einer Distanzfunktion bekannt ist, jedoch auch strengere Ähnlichkeitsmaße aus einem Distanzmaß herleiten, denn $1/1+d$ kann niemals 0 werden. Wenn, wie in Abschnitt 4.4.2 beschrieben wird, ein Ähnlichkeitswert von 0 als Abbruchkriterium innerhalb eines Algorithmus zur Ähnlichkeitsberechnung zweier Prozessmodelle verwendet wird, muss ein Ähnlichkeitswert von 0 zumindest prinzipiell erreicht werden können. Deshalb sollte die angegebene Transformation eines Distanzmaßes in ein Ähnlichkeitsmaß nur dann verwendet werden, wenn keine andere Transformation möglich ist.

2.1.2 Gleichheit von Prozessmodellen

Neben der Bestimmung der Ähnlichkeit von Prozessmodellen stellt sich auch oft die Frage, wann zwei Prozessmodelle wirklich gleich sind. Sind zwei Prozessmodelle, die genau den gleichen Ablauf beschreiben, jedoch in unterschiedlichen Sprachen notiert sind, gleich? Auch innerhalb ein und derselben Modellierungssprache lassen sich oftmals die haargenau gleichen Abläufe auf unterschiedliche Art und Weise modellieren, siehe die beiden Beispielmodelle in Abbildung 2.1. Unter anderem Hidders et al. (2005) haben sich mit diesen Fragen beschäftigt. Von Hidders et al. (2005) werden zwei Prozessmodelle als gleich angesehen, wenn die Mengen ihrer Ausführungspfade übereinstimmen (vgl. Li et al., 2008). Unter dieser Sichtweise beeinflussen alternative Darstellungsweisen die Gleichheit also nicht. Hierfür müssen sich entsprechende Aktivitäten in den Prozessmodellen bekannt sein, außerdem muss es zu jeder Aktivität genau eine Entsprechung im anderen Modell geben. Wie Hidders et al. (2005) im Schluss ihrer Arbeit selbst sagen, gibt ihre Arbeit noch keine definitive Antwort auf die Frage nach der Gleichheit. Für die vorliegende Arbeit soll deswegen folgender Gleichheitsbegriff gelten: Zwei Prozessmodelle sind *unter Verwendung eines bestimmten Ähnlichkeitsmaßes* gleich, wenn dieses Maß einen Ähnlichkeitswert von 1 ergibt, was dem maximal möglichen Wert gemäß Definition 2.1 und Gleichung (2.1) entspricht. Verschiedene Ähnlichkeitsmaße können hierbei natürlich zu unterschiedlichen Ergebnissen kommen. Die von Hidders et al. (2005) angeführte Gleichheitsaussage trifft also dann zu, wenn die Gleichheit der Ausführungspfade als Ähnlichkeitsmaß formuliert werden kann und dieses Maß zur Messung der Ähnlichkeit akzeptiert ist. In Abschnitt 2.2.4.1 ist ein solches Ähnlichkeitsmaß beschrieben.

[3] Erfüllt *sim* die Eigenschaft $sim(A,B) = 1 \Rightarrow A = B$ nicht, so tut dies das daraus gebildete Distanzmaß ebenfalls nicht. Die Eigenschaft $d(A,B) = 0 \Rightarrow A = B$ gilt dann nicht.

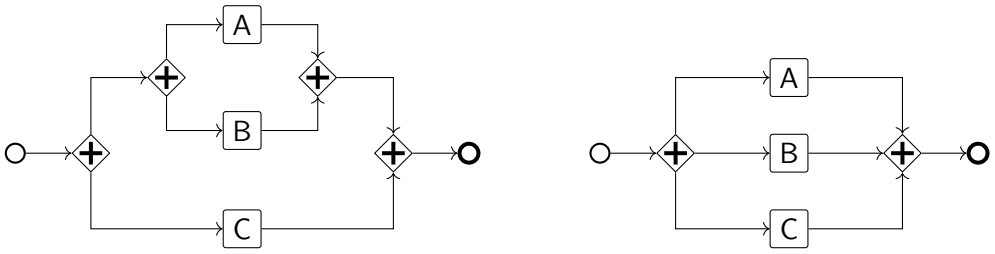


Abbildung 2.1: Zwei unterschiedliche Prozessmodelle mit jeweils der gleichen Menge an Ausführungspfaden.

2.2 Ähnlichkeitsabgleich imperativer Prozessmodelle mit Ähnlichkeitsmaßen auf Basis von Abbildungen

Nachdem der Ähnlichkeitsbegriff, wie er in dieser Arbeit zu verstehen ist, definiert ist, werden in diesem Abschnitt Ähnlichkeitsmaße aus verwandten Arbeiten vorgestellt, die vielfach mit eigenen Beispielen veranschaulicht werden. In der Literatur existiert eine breite Auswahl an Verfahren zur Berechnung der Ähnlichkeit von Prozessmodellen, wobei diese (fast) alle auf imperativ modellierte Prozesse ausgelegt sind. Die vorgestellten, gängigsten Methoden aus der Literatur werden vor allem auch in Hinblick auf eine mögliche Erweiterbarkeit bzw. Übertragbarkeit, wie sie in Abschnitt 1.4 angesprochen wird, betrachtet.

Zunächst wird das grundsätzliche, vierstufige Vorgehen, das auf dem Editierabstand von Graphen beruht, gezeigt (Abschnitt 2.2.1) und anschließend auf Verfahren auf Label-, Struktur- und Verhaltensbasis (Abschnitte 2.2.2, 2.2.3 und 2.2.4) eingegangen, die bei Schritt 2 des vierstufigen Verfahrens angewendet werden können. Das vierstufige Verfahren stellt, implizit oder explizit genannt, in vielen verwandten Arbeiten die Grundlage der Ähnlichkeitsmessung von Prozessmodellen dar. Korrespondenzen in den verglichenen Modellen werden dabei mit Hilfe einer Abbildung der Modellelemente aufgezeigt. Einige Methoden, die keine Abbildung, und somit auch nicht das vierstufige Verfahren, zugrunde legen, werden in Abschnitt 2.3 vorgestellt, bevor eine Zusammenfassung und Einordnung der Methoden aus der Literatur in Abschnitt 2.4 gegeben wird.

2.2.1 Grundsätzliches, vierstufiges Vorgehen

Eine gängige Art und Weise, Prozessmodelle miteinander zu vergleichen, ist die, vor dem Berechnen der Ähnlichkeit in einem ersten Schritt eine Abbildung zwischen den Elementen der zu vergleichenden Modelle festzulegen (vgl. Dijkman et al., 2009b). Anhand dieser Abbildung wird dann eine Ähnlichkeit der einander zugeordneten Prozessmodellelemente berechnet. Dieser Wert hängt von der zugrunde gelegten Abbildung ab und kann über eine optimale Wahl der Abbildung maximiert werden. So erhält man nicht nur einen Ähnlichkeitswert, sondern gleichzeitig auch Modellkorrespondenzen sowie Hinweise, welche Teile der verglichenen Prozessmodelle sich stark voneinander unterscheiden. Dieser Ansatz stammt ursprünglich aus dem Gebiet des Graph Matchings und verwendet Konzepte zur Berechnung des Editierabstands von Graphen (*graph edit distance*; GED) (Bunke und Jiang, 2000). Der Ansatz wird, in angepasster Form, auch in Kapitel 4 übernommen. Abbildung 2.2 illustriert das vierstufige Vorgehen.

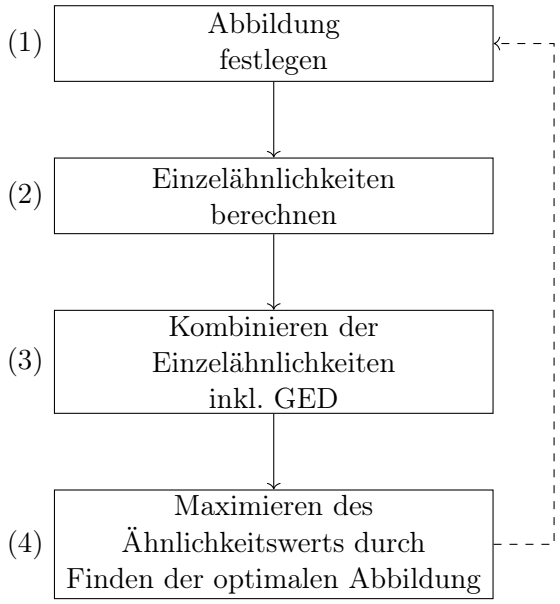


Abbildung 2.2: Vierstufiges Vorgehen zum Ähnlichkeitsabgleich.

2.2.1.1 1:1-Abbildung zwischen zwei Prozessmodellen

Die Abbildung zwischen zwei Prozessmodellen erfolgt, übernommen von Dijkman et al. (2009a), über eine Zuordnung der einzelnen Knoten gemäß der folgenden Definition:

Definition 2.2 (1:1-Abbildung)**.** Seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle. Eine Abbildung $M : G_1 \to G_2$ ist gegeben durch

$$M : N_1 \rightarrowtail N_2,$$

wobei M eine partiell injektive Funktion[4] ($\rightarrowtail$) ist, die Knoten aus G_1 Knoten aus G_2 zuordnet. Mit M^{-1} wird die Umkehrabbildung von M bezeichnet, die wiederum partiell injektiv ist.

Alternativ ist es auch möglich, die Funktion $M : G_1 \to G_2$ als partiell injektive Funktion $M : A_1 \rightarrowtail A_2$ auf den Aktivitätenmengen zu definieren. Eine beispielhafte Abbildung zwischen zwei Modellen ist in Abbildung 2.3 gezeigt. Eine solche Abbildung ist Voraussetzung für den nächsten Schritt, der Ähnlichkeiten für die Knotenpaare der Abbildung bestimmt.

2.2.1.2 Fallunterscheidung: Paarweise Ähnlichkeit der Abbildungselemente oder gemeinsamer Ähnlichkeitswert

Ist eine Abbildung zwischen zwei zu vergleichenden Prozessmodellen definiert, so gibt es nun zwei Möglichkeiten, wie weiter vorgegangen werden kann: Entweder werden zunächst die Elemente der Abbildung, also die einzelnen Knotenpaare, miteinander verglichen und Ähnlichkeitswerte gefunden, die im Anschluss zu einem gemeinsamen Ähnlichkeitswert zusammengerechnet werden, oder es wird gleich ein solcher gemeinsamer Ähnlichkeitswert berechnet, ohne die Ähnlichkeiten der einzelnen Paare explizit zu betrachten. Beide Varianten haben gemein,

[4]Eine Funktion $M : N_1 \rightarrowtail N_2$ heißt dann partiell injektiv, wenn für ihre Definitionsmenge $D(M)$ gilt, dass $D(M) \subset N_1$, und M auf $D(M)$ injektiv ist.

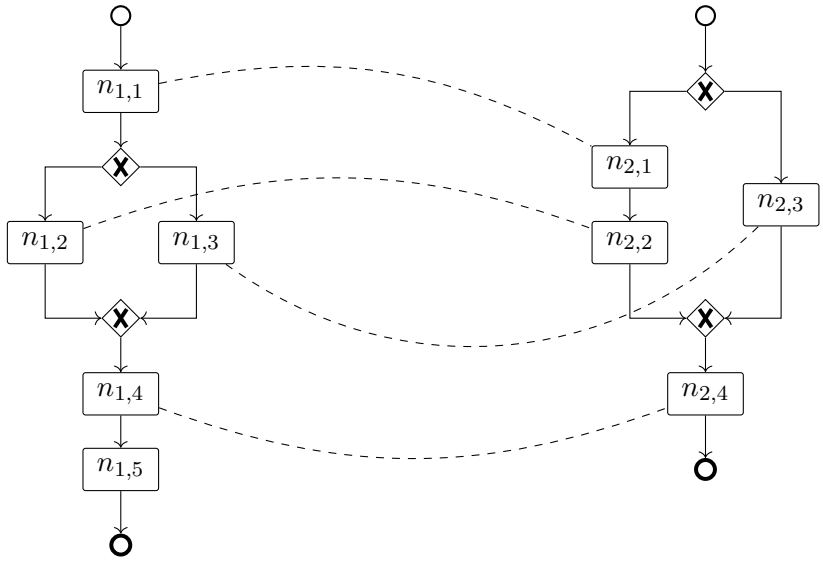


Abbildung 2.3: Beispielabbildung mit einer 1:1-Zuordnung von Aktivitäten, angedeutet durch gestrichelte Linien.

dass zum Schluss ein gemeinsamer Ähnlichkeitswert auf Grundlage einer Abbildung gefunden wird, wobei dies bei der ersten Variante über einen Zwischenschritt – die Betrachtung der einzelnen Knotenpaare – erreicht wird. Selbstverständlich können diese Konzepte für unterschiedliche Aspekte der betrachteten Prozessmodelle, wie den Aktivitätenbeschreibungen und der Ablauflogik, auch kombiniert werden. Ob paarweise oder gemeinsame Ähnlichkeitswerte verwendet werden, hängt dabei einzig und allein von der Wahl des Ähnlichkeitsmaßes ab, das entweder auf Knotenmengen oder auf kompletten Prozessmodellen definiert sein kann. Im weiteren Verlauf werden Beispiele für beide Arten gezeigt. Die Auswahl der Ähnlichkeitsmaße ist für den Abgleich elementar, wobei genauere Diskussionen hierzu bei der Definition des jeweiligen Maßes erfolgen.

Indirekte gemeinsame Ähnlichkeit über Ähnlichkeit der Knotenpaare Für die gegebene Abbildung bzw. für ihre Elemente $(n_1, n_2) \in M$ mit $n_1 \in N_1$ und $n_2 \in N_2$ und $M : G_1 = (N_1, E_1, \lambda_1) \rightarrow G_2 = (N_2, E_2, \lambda_2)$ wird, unter Anwendung eines geeigneten Ähnlichkeitsmaßes, die Ähnlichkeit paarweise berechnet. Mögliche Ähnlichkeitsberechnungen, die sich auf einzelne Knotenpaare beschränken, werden in Abschnitt 2.2.2 genauer vorgestellt. Sie beruhen meist auf einem Vergleich von Aktivitätenbeschreibungen. Allgemein ist die Ähnlichkeit auf den Knoten eines Prozessmodells in folgender Weise definiert:

Definition 2.3 (Knotenähnlichkeit)**.** Eine Funktion *sim* ist ein Knotenähnlichkeitsmaß für zwei Prozessmodelle G_1 und G_2, falls

$$sim : N_1 \times N_2 \rightarrow [0, 1],\ sim(n_1, n_2) \in [0, 1]\ \forall\ n_1 \in N_1, n_2 \in N_2$$

und *sim* die Eigenschaften aus Definition 2.1 erfüllt.

Um die Ergebnisse solcher Ähnlichkeitsberechnungen stärker zu kontrastieren, kann zusätzlich ein Schwellenwert $k \in (0, 1)$ eingeführt werden, sodass Ähnlichkeitswerte, die diesen

Schwellenwert unterschreiten, nicht weiter berücksichtigt werden (Dijkman et al., 2011). Ähnlichkeiten kleiner als k werden auf null gesetzt:

$$sim_k(n_1, n_2) = \begin{cases} sim(n_1, n_2), & \text{falls } sim(n_1, n_2) \geq k \\ 0, & \text{sonst} \end{cases} \tag{2.3}$$

Aus der Ähnlichkeit der Knotenpaare erhält man den gemeinsamen Ähnlichkeitswert, $fsim_M$ genannt, über eine Mittelung der Einzelähnlichkeiten. Für die Mittelung wird die Anzahl der mittels der Abbildung M abgebildeten Knoten benötigt.

Definition 2.4 (Abgebildete und gelöschte Knoten)**.** Für eine Abbildung M gemäß Definition 2.2 und zwei Prozessmodelle G_1 und G_2 ist die Menge

$$subn_M := \{n \in N_1 \cup N_2 \mid (n, k_2) \in M \vee (k_1, n) \in M,\ k_1 \in N_1, k_2 \in N_2\}$$

die Menge aller abgebildeten Knoten. Dementsprechend bezeichnet

$$skipn_M := (N_1 \cup N_2) \setminus subn_M$$

die Menge aller gelöschten Knoten.

Die Menge $skipn_M$ wird in den nachfolgenden Formeln in dieser Arbeit nicht verwendet, da die Notation, wie in der Anmerkung am Ende von Abschnitt 2.2.1.3 erwähnt, auf eine einheitliche Form angepasst ist. In der Literatur (z. B. von Dijkman et al., 2009b) taucht $skipn_M$ dagegen in weiteren Formeln auf, weshalb die Menge hier der Vollständigkeit halber definiert ist. Damit kann nun die mittlere Ähnlichkeit, wie unter anderem von Dijkman et al. (2009a) übernommen, definiert werden:

Definition 2.5 (Mittlere Ähnlichkeit der abgebildeten Knoten)**.** Seien G_1 und G_2 zwei Prozessmodelle, die mit Abbildung $M : G_1 \to G_2$ aufeinander abgebildet werden, und sei sim ein Ähnlichkeitsmaß auf den Knotenpaaren der Abbildung. Dann ist die mittlere Ähnlichkeit der beiden Modelle auf Basis von sim gegeben durch

$$fsim_M := \frac{2 \sum_{(n_1,n_2)\in M} sim(n_1, n_2)}{|subn_M|},$$

falls $|M| \geq 1$. Falls M die leere Abbildung ist, so ist $fsim_M = 0$.

Das Multiplizieren mit 2 im Zähler ist notwendig, da in $|subn_M|$ die abgebildeten Knoten aus beiden Prozessmodellen einzeln zusammengezählt werden, in sim jedoch immer nur Knotenpaare stehen. Der Wert von $|subn_M|$ kann maximal $|N_1| + |N_2|$ groß werden, der Wert von $\sum_{(n_1,n_2)\in M} sim(n_1, n_2)$ maximal $\min\{|N_1|, |N_2|\}$. Um also für $fsim_M$ einen Wert von 1 überhaupt möglich zu machen, wird die Anzahl der betrachteten Elemente in Zähler und Nenner durch Malnehmen mit 2 im Zähler angeglichen.

Dijkman et al. (2009a) oder auch La Rosa et al. (2010) definieren an dieser Stelle in ihrer Arbeit keine mittlere Ähnlichkeit, sondern eine mittlere Unähnlichkeit, indem im Zähler der Summe in Definition 2.5 statt $sim(n_1, n_2)$ steht: $(1 - sim(n_1, n_2))$. Wie in der Anmerkung in Abschnitt 2.2.1.3 sind diese Zwischenwerte ineinander umrechenbar und führen am Ende zum selben Ergebnis. Aus Gründen der Konsistenz zu den später eingeführten, neuen Konzepten, ist hier die umgeformte Definition angeführt.

Direkte gemeinsame Ähnlichkeit Einige Abgleichsmethoden betrachten nicht die einzelnen Knotenpaare und deren Ähnlichkeit, sondern leiten einen Ähnlichkeitswert direkt aus der gesamten Abbildung her. Für diese Methoden werden vor allem in Abschnitt 2.2.4 über verhaltensbasierte Ähnlichkeitsbestimmung einige Beispiele vorgestellt. Auch hier ist das Ziel, einen Ähnlichkeitswert $fsim_M \in [0,1]$ herzuleiten, jedoch ohne dass dafür eine Ähnlichkeitsfunktion auf Knotenpaaren sim benötigt wird.

2.2.1.3 Von M induzierter globaler Ähnlichkeitswert unter Berücksichtigung gelöschter Modellelemente

Ist ein Ähnlichkeitswert $fsim_M$ auf Grundlage der Abbildung M gefunden, so wird in einem nächsten Schritt dieser Ähnlichkeitswert mit den unter M nicht abgebildeten, d. h. gelöschten Knoten und Kanten verrechnet, um eine „vernünftige“ Abbildung zu gewährleisten. Der Begriff der „vernünftigen“ Abbildung wird im Folgenden näher erläutert. Zunächst werden, neben den abgebildeten und gelöschten Knoten, die in Definition 2.4 definiert werden, auch abgebildete und gelöschte Kanten definiert, wobei diese Definitionen analog zueinander sind. Eine Kante eines Modells wird dann unter M abgebildet, wenn die beiden Knoten, durch die sie definiert ist, im Bild von M und damit im zweiten Modell ebenfalls eine Kante bilden.

Definition 2.6 (Abgebildete und gelöschte Kanten)**.** Gegeben sei eine Abbildung M zwischen G_1 und G_2. Alle Kanten $(n_1, m_1) \in E_1$ aus G_1, für die gilt, dass $\exists (n_2, m_2) \in E_2$ und die Kanten $(n_2, m_2) \in E_2$, für die gilt dass $\exists (n_1, m_1) \in E_1$ mit $M(n_1) = n_2$ und $M(m_1) = m_2$, werden als abgebildete Kanten bezeichnet:

$$\begin{aligned} sube_M := \{(n,m) \in E_1 \cup E_2 \mid \exists (n_1, m_1) \in E_1 : (M(n_1), M(m_1)) = (n,m) \\ \vee \exists (n_2, m_2) \in E_2 : (M(n), M(m)) = (n_2, m_2)\} \end{aligned}$$

Alle anderen Kanten werden gelöscht und in der Menge $skipe_M$ zusammengefasst:

$$skipe_M := (E_1 \cup E_2) \setminus sube_M$$

Die Ähnlichkeit zweier Modelle anhand der gegebenen Abbildung wird, wie eingangs erwähnt, nicht nur aus den Ähnlichkeitswerten der abgebildeten Elemente, also aus $fsim_M$, berechnet, sondern bezieht auch den Anteil der abgebildeten Knoten und Kanten mit ein, um Abbildungen zu bestrafen, die kaum Korrespondenzen angeben, also im Vergleich zur Größe der Prozessmodelle nur wenige Knotenpaare umfassen. Hierzu wird der Anteil der abgebildeten Knoten und Kanten im Vergleich zu allen Knoten bzw. Kanten bestimmt. Die Definition ist wiederum von Dijkman et al. (2009a) bzw.La Rosa et al. (2010) übernommen.

Definition 2.7 (Anteil abgebildeter Knoten und Kanten)**.** Es seien G_1 und G_2 zwei Prozessmodelle, die mit der Abbildung $M : G_1 \to G_2$ aufeinander abgebildet werden. Die absoluten Anzahlen der abgebildeten (gelöschten) Knoten bzw. Kanten sind mit $subn_M$ ($skipn_M$) bzw. $sube_M$ ($skipe_M$) bezeichnet. Der Anteil der abgebildeten Knoten ist dann definiert über

$$fsubn_M := \frac{|subn_M|}{|N_1| + |N_2|} = 1 - \frac{|skipn_M|}{|N_1| + |N_2|} =: 1 - fskipn_M$$

und der Anteil der abgebildeten Kanten über

$$fsube_M := \frac{|sube_M|}{|E_1| + |E_2|} = 1 - \frac{|skipe_M|}{|E_1| + |E_2|} =: 1 - fskipe_M.$$

Auch $fsubn_M$ und $fsube_M$ nehmen wie $fsim_M$ Werte im Intervall $[0, 1]$ an. Die Anteile $fsubn_M$ und $fsube_M$ sind nahe bei null, wenn die Abbildung M nur einen geringen Anteil der in den Modellen vorhandenen Knoten und Kanten überhaupt berücksichtigt. Tendenziell ist es so, dass je selektiver die Abbildung M ist, desto besser, d. h. näher bei eins, kann der Wert von $fsim_M$ werden. Dies führt jedoch eher zu schlechteren Werten von $fsubn_M$ und $fsube_M$. Um diese Effekte zu berücksichtigen, werden die drei Anteile gemittelt, um einen globalen Abgleich zu erhalten. Diese Mittelung erfolgt über eine Gewichtung der drei Anteile.

Definition 2.8 (Von M induzierte globale Ähnlichkeit)**.** Für zwei Modelle G_1 und G_2 und eine Abbildung M ist die von M induzierte globale Ähnlichkeit, $gsim_M$, gegeben durch

$$gsim_M(G_1, G_2) = wsubn \cdot fsubn_M + wsube \cdot fsube_M + wsim \cdot fsim_M,$$

wobei $wsubn + wsube + wsim = 1$ und $wsubn, wsube, wsim \geq 0$ gelte.

Diese von M induzierte globale Ähnlichkeit nimmt wieder Werte in $[0, 1]$ an und ist bei ähnlichen Modellen und einer geschickt gewählten Abbildung M nahe eins, während der Vergleich unähnlicher Modelle Werte nahe null ausgibt. Da der so gewonnene Ähnlicheitswert natürlich von der Abbildung M abhängt und diese Abbildung gerade die korrespondierenden Elemente beider Modelle anzeigt, muss in einem letzten Schritt noch die beste Abbildung gefunden werden. Diese gibt dann den besten Ausgleich zwischen größtmöglicher Ähnlichkeit der abgebildeten Knoten und den Anteilen der überhaupt berücksichtigten Knoten und Kanten wieder. Der Begriff „beste Abbildung“ meint dabei diejenige Abbildung, unter der der Ähnlichkeitswert $gsim_M$ maximiert wird. Die Ausprägung der Gewichte ist hierbei ebenfalls von entscheidender Bedeutung. Eine neutrale Möglichkeit ist es, die drei Summanden gleich stark zu gewichten. Ansonsten können die Gewichte den zu vergleichenden Prozessmodellen angepasst werden. Falls Benchmarks existieren, können sie auch dahingehend gesetzt werden.

Anmerkung In den verwandten Arbeiten, in denen derselbe Ansatz verwendet wird, beispielsweise von Dijkman et al. (2009b) oder von La Rosa et al. (2010), wird nicht $fsim_M$ definiert, sondern $fdist_M$ als

$$fdist_M = \frac{2 \sum_{(n_1,n_2) \in M} (1 - sim(n_1, n_2))}{|subn_M|}.$$

Dieses $fdist_M$ ist quasi eine normierte Unähnlichkeit. Entsprechend wird dann auch bei der Bestimmung der von M induzierten globalen Ähnlichkeit wie folgt vorgegangen:

$$gsim_M(G_1, G_2) = 1 - (wskipn \cdot fskipn_M + wskipe \cdot fskipe_M + wdist \cdot fdist_M)$$

mit $wskipn + wskipe + wdist = 1$ und $wskipn, wskipe, wdist \geq 0$. Dieser von M induzierte Ähnlichkeitswert ist jedoch gleich zu dem, der in Definition 2.8 eingeführt wurde, wie die folgende Rechnung zeigt.

Es gilt, dass

$$|M| = |\{(n_1, n_2) \mid (n_1, n_2) \in M\}| = \frac{|subn_M|}{2} \Leftrightarrow 2|M| = |subn_M|.$$

Damit ist

$$\begin{aligned}
1 - fsim_M &= 1 - \frac{2\sum_{(n_1,n_2)\in M} sim(n_1,n_2)}{|subn_M|} \\
&= \frac{|subn| - 2\sum_{(n_1,n_2)\in M} sim(n_1,n_2)}{|subn_M|} \\
&= \frac{2|M| - 2\sum_{(n_1,n_2)\in M} sim(n_1,n_2)}{|subn_M|} \\
&= \frac{2\left(|M| - \sum_{(n_1,n_2)\in M} sim(n_1,n_2)\right)}{|subn_M|} \\
&= \frac{2\sum_{(n_1,n_2)\in M}(1 - sim(n_1,n_2))}{|subn_M|} \\
&= fdist_M. \\
\Rightarrow fsim_M &= 1 - fdist_M.
\end{aligned}$$

Eingesetzt in die alternative Definition von $gsim_M$ ergibt sich

$$\begin{aligned}
& 1 - (wskipn \cdot fskipn_M + wskipe \cdot fskipe_M + wdist \cdot fdist_M) \\
&= 1 - (wskipn(1 - fsubn_M) + wskipe(1 - fsube_M) + wdist \cdot (1 - fsim_M)) \\
&= 1 - wskipn + wskipn \cdot fsubn_M - wskipe + wskipe \cdot fsube_M - wdist + wdist \cdot fsim_M \\
&= 1 - (wskipn + wskipe + wdist) + wskipn \cdot fsubn_M + wskipe \cdot fsube_M + wdist \cdot fsim_M \\
&= wskipn \cdot fsubn_M + wskipe \cdot fsube_M + wdist \cdot fsim_M.
\end{aligned}$$

Die Berechnungen liefern also, mit Umbenennung des Gewichts $wsim$ in $wdist$ denselben Ähnlichkeitswert.

Die Ähnlichkeitsberechnung anhand der in Definition 2.8 gegebenen Anweisung erlaubt es, auch solche Ähnlichkeitswerte direkt als $fsim_M$ in die Berechnung von $gsim_M$ einzubeziehen, die nicht erst über einzelne Knotenpaare bestimmt werden, sondern direkt aus den kompletten Prozessmodellen. Deswegen wird in dieser Arbeit der direkte Weg über $fsim_M$ und nicht der über $fdist_M$ verwendet.

2.2.1.4 Maximierung des Ähnlichkeitswerts durch Finden einer optimalen Abbildung

Dieser letzte Schritt des Ähnlichkeitsabgleichs ist eigentlich eine Wiederholung der bisher durchgeführten Berechnungen und zielt darauf ab, eine optimale Abbildung zwischen den beiden zu vergleichenden Prozessmodellen zu finden, wobei diese optimale Abbildung den globalen Ähnlichkeitswert $gsim_M$ maximiert.

Definition 2.9 (Globale Ähnlichkeit)**.** Die globale Ähnlichkeit $gsim$ der beiden Prozessmodelle G_1 und G_2 erhält man über

$$gsim(G_1, G_2) = \max_M gsim_M(G_1, G_2).$$

Hierbei ist die optimale Abbilung M^* gegeben durch

$$M^* = \operatorname*{argmax}_{M:G_1 \to G_2} gsim_M(G_1, G_2).$$

Es ist dann $gsim(G_1, G_2) = gsim_{M^*}(G_1, G_2)$. Zur Implementierung dieses Optimierungs- bzw. Maximierungsproblems werden Algorithmen wie Greedy oder A*-Algorithmen verwendet (siehe z. B. Dijkman et al., 2009a), da das Problem M^* zu finden von exponentieller Ordnung ist. Beachte, dass M^* nach obiger Definition nicht eindeutig sein muss. Grundsätzlich kann argmax eine Menge an Funktionen ausgeben. Ist die Mächtigkeit der Lösungsmenge echt größer eins, dann bestimme als optimale Abbildung ein beliebiges Element dieser Menge.

Mit Hilfe eines solchen Optimierungsalgorithmus ist es nun möglich, auf effiziente Weise einen Ähnlichkeitsabgleich zwischen zwei Prozessmodellen durchzuführen. Dabei gibt es verschiedene Möglichkeiten, wie die Ähnlichkeit zwischen je zwei Knoten, die im in Abschnitt 2.2.1.2 beschriebenen Schritt des mehrstufigen Ansatzes benötigt wird, definiert sein kann. Einen vergleichenden Überblick über verschiedene Methoden bieten beispielsweise Becker und Laue (2012) und Starlinger et al. (2014) oder die Arbeiten, die aus den beiden Process Model Matching Contests 2013 (Cayoglu et al., 2013) und 2015 (Antunes et al., 2015) hervorgegangen sind. Auch Thaler et al. (2017) vergleichen verschiedene Methoden und betrachten sie vor allem hinsichtlich ihrer Korrelation untereinander und ihrer Implementierung und der damit verbundenen Performanz.

2.2.2 Labelbasierte Ansätze zum Ähnlichkeitsabgleich

Ein naheliegender Ansatz zum Bestimmen der Ähnlichkeit zweier Knoten ist, sich die Beschriftungen der Knoten anzuschauen. Die einfachste Herangehensweise ist die, die Zeichenketten ohne eine weitergehende Interpretation zu betrachten (syntaktische Ähnlichkeit). Um die Bedeutung der Beschriftungen mit zu berücksichtigen, muss die Semantik der Labels zum Ähnlichkeitsabgleich hinzugezogen werden. In den im Folgenden vorgestellten Ähnlichkeitsmaßen wird die Reihenfolge der Knoten grundsätzlich nicht beachtet (Becker und Laue, 2012). Verwendet man die Maße aber in Zusammenhang mit dem mehrstufigen Ähnlichkeitsabgleich aus Abschnitt 2.2.1, dann wird dieser vermeintliche Nachteil durch das Hinzuziehen der abgebildeten Kanten bei der Berechnung der Ähnlichkeit ausgeglichen.

In den Abschnitten 2.2.2.1 bis 2.2.2.3 werden verschiedene Maße betrachtet, die auf den Beschreibungen als Strings arbeiten. Sie haben den Vorteil, dass sie relativ einfach aufgebaut sind und kleinere Rechtschreibfehler nicht zu sehr bestrafen. Semantische Methoden (Abschnitt 2.2.2.4) erfordern einen größeren Aufwand, berücksichtigen jedoch die den Beschreibungen zugedachten Bedeutungen. Rechtschreibfehler können sich jedoch kritisch auf die Ergebnisse auswirken. In den Abschnitten 2.2.2.5 bis 2.2.2.7 werden Ähnlichkeiten auf Wortebene statt auf Ebene der einzelnen Zeichen bestimmt. Der Aufwand dieser Methoden steht tendenziell zwischen dem der syntaktischen und dem der semantischen Methoden. Alle vorgestellten Methoden bestimmen die Ähnlichkeit von Aktivitätenpaaren (Fall 1 in Abschnitt 2.2.1.2).

2.2.2.1 Syntaktische Ähnlichkeit

Eine Möglichkeit, wie Prozessmodelle anhand ihrer Beschriftung abgeglichen werden können, ist die Berechnung der sogenannten Levenshtein-Distanz bzw. Editierdistanz (*string-edit distance*; sed) zweier Zeichenketten, wie sie z. B. von Dijkman et al. (2009b) vorgeschlagen wird. Bei der Levenshtein-Distanz zwischen zwei Zeichenketten s_1 und s_2 wird gezählt, wie viele atomare Operationen mindestens benötigt werden, um s_1 in s_2 zu überführen:

$$sed(s_1, s_2) = \min(|\text{atomare Operationen für den Übergang von } s_1 \text{ nach } s_2|) \in \mathbb{N}_0.$$

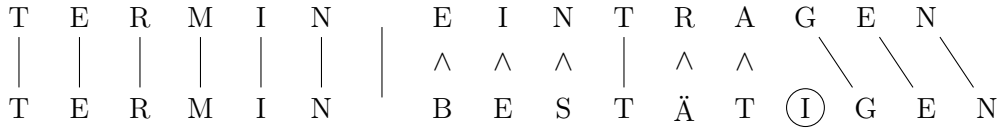


Abbildung 2.4: Beispiel für String-Edit-Similarity.

Atomare Umformungsoperatoren sind hierbei das Einfügen, das Löschen sowie das Ersetzen eines einzelnen Zeichens. Sind die beiden Zeichenketten s_1 und s_2 die gleichen, so ist die Distanz null. Aus der Levenshtein-Distanz lässt sich leicht eine normierte Levenshtein-Ähnlichkeit ableiten. Da für die Levenshtein-Distanz gilt, dass sie höchstens so groß ist wie die Länge der längeren Zeichenkette, kann diese Länge zur Normierung verwendet werden. Um einer geringen Distanz eine große Ähnlichkeit und umgekehrt einer großen Distanz eine geringe Ähnlichkeit zuzuweisen, wird der normierte Wert dann noch mittels „1−normierte Distanz“ zu einer normierten Ähnlichkeit überführt. Es ergibt sich somit für die Levenshtein-Ähnlichkeit zweier Zeichenketten s_1 und s_2:

$$sim_{sed}(s_1, s_2) = 1 - \frac{sed(s_1, s_2)}{\max(|s_1|, |s_2|)} \in [0, 1],$$

wobei mit $|s_i|$ die Länge der Zeichenkette s_i bezeichnet wird. Sind s_1 und s_2 gleich, so ist die Ähnlichkeit eins. Klar ist bei dieser Ähnlichkeitsberechnung ebenfalls, dass unabhängig von den verwendeten Zeichen eine unterschiedliche Stringlänge auf jeden Fall zu einer Minderung des Ähnlichkeitswerts führt. Ist der längere der beiden Strings l Zeichen lang und der Unterschied der Stringlängen beträgt $u < l$, so ist die Levenshtein-Ähnlichkeit maximal $1 - u/l < 1$.

Bezogen auf Prozessmodelle sind die zu vergleichenden Strings die Beschriftungen der Knoten, die über die Funktion λ gegeben sind: $s_i = \lambda(n_i)$. Die Ähnlichkeit zweier Knoten n_1 und n_2 kann somit mit der Ähnlichkeit der Beschriftungen gleichgesetzt werden: $sim_{sed}(n_1, n_2) = sim_{sed}(\lambda(n_1), \lambda(n_2))$.

Beispiel 2.1. Für die beiden Zeichenketten s_1 =„Termin eintragen“ und s_2 =„Termin bestätigen“ ist $16 = |s_1| < |s_2| = 17$ und die Editierdistanz beträgt $sed(s_1, s_2) = 6$, wie Abbildung 2.4 zeigt. Fünf Zeichen müssen ersetzt werden (∧) und ein Zeichen hinzugefügt bzw. gelöscht werden (◯). Dies ergibt eine Levenshtein-Ähnlichkeit von

$$sim_{sed}(s_1, s_2) = 1 - \frac{6}{17} = \frac{11}{17} \approx 0,65$$

Ebenfalls unter Verwendung der Levenshtein-Distanz schlagen Maedche und Staab (2002) ein alternatives Maß vor:

$$\begin{aligned} sim_{sed'}(s_1, s_2) &= \max\left(0, \frac{\min(|s_1|, |s_2|) - sed(s_1, s_2)}{\min(|s_1|, |s_2|}\right) \\ &= \max\left(0, 1 - \frac{sed(s_1, s_2)}{\min(|s_1|, |s_2|}\right) \in [0, 1] \end{aligned} \tag{2.4}$$

Da $\min(|s_1|, |s_2|) \leq \max(|s_1|, |s_2|)$ gilt, ist $sim_{sed'}(s_1, s_2) \leq sim_{sed}(s_1, s_2)$. Das Maß $sim_{sed'}$ kann also als strenger als sim_{sed} aufgefasst werden. Die Maße sim_{sed} und $sim_{sed'}$,

die eine Ähnlichkeit von Knotenpaaren berechnen, genügen Definition 2.3 und können somit beide für die Funktion sim in der Definition von $fsim_M$ eingesetzt werden. Wie in Abschnitt 4.1 beschrieben, ist es relativ einfach, diese syntaktischen Ähnlichkeitsmaße auf M:N-Abbildungen anzupassen. Zudem existieren viele Implementierungen der Levenshtein-Distanz, was eine Anwendung einfach macht. Kleine Rechtschreibfehler, z. B. ein vergessenes oder ein falsches Zeichen, wirken sich insoweit auf das Ergebnis aus, als dass entsprechend viele atomare Operationen bei der Bestimmung der Levenshtein-Distanz hinzukommen. Je länger der betrachtete String, desto geringer ist die relative Wirkung eines Fehlers. Homonyme werden mit einer Ähnlichkeit von 1 bewertet. Die Bedeutung der Beschreibungen wird nicht erfasst. Bei Prozessmodellen, deren Aktivitäten mit einem standardisierten Vokabular und Stil (z. B. Nominalstil) formuliert sind, lassen sich gute Ergebnisse erzielen (siehe z. B. Cheatham und Hitzler, 2013).

2.2.2.2 Syntaktische Ähnlichkeit mit Stemming

Wie von Dijkman et al. (2009b, 2011) vorgeschlagen, kann man die Knotenbeschreibungen zunächst einer Vorverarbeitung unterziehen, um kleinere Abweichungen der Beschriftung, die keinen Einfluss auf die Güte der Abbildung haben, auszuschließen. Die Beschriftung wird in einem ersten Schritt in eine Liste an Wörtern überführt. So werden besondere Symbole wie beispielsweise Zeilenumbrüche entfernt. Alle Zeichen werden als Kleinbuchstaben geschrieben. Sogenannte Stoppwörter, also häufig verwendete Wörter wie bestimmte und unbestimmte Artikel („der“, „die“, „das“, „einer“, „eine“ usw.), Konjunktionen („und“, „oder“, „doch“ usw.) oder auch häufig verwendete Präpositionen („an“, „in“, „von“ usw.) werden aus der Liste entfernt genau wie sonstige Füllwörter. Das Streichen von doppelten Wörtern und ein Sortieren der übrig gebliebenen Wörter kann ebenfalls erfolgen. Diese Wörter werden, sofern möglich, auf ihren Wortstamm zurückgeführt. Möglichkeiten hierfür bieten Lovins' oder Porters Stemming Algorithmen (Lovins, 1968; Porter, 1980). Die so vereinfachte Wortliste wird in einen String zurückgeführt, um beispielsweise die Levenshtein-Ähnlichkeit darauf anwenden zu können. Gerade der Einsatz von Stemming ist im Falle von M:N-Abbildungen, wie in Abschnitt 4.1 erläutert, als durchaus sinnvoll zu erachten. Der Einsatz von Stemming kann die Abgleichsergebnisse deutlich verbessern (Cheatham und Hitzler, 2013), stellt aber zusätzlichen Aufwand dar. Die Güte von Stemming-Algorithmen hängt unter anderem stark von der verwendeten Sprache ab (Paice, 1994). Für das Englische gibt es beispielsweise gut funktionierende Implementierungen, während diese für die deutsche Sprache oft nicht korrekt arbeiten oder einen deutlich höheren Aufwand erfordern (Braschler und Ripplinger, 2004).

2.2.2.3 Weitere Stringmetriken

In der Literatur werden noch eine Reihe weiterer Stringmetriken genannt, die wie die Levenshtein-Ähnlichkeit eine rein syntaktische Betrachtung der Knotenbeschreibungen vornehmen. Einige werden nun der Vollständigkeit halber kurz vorgestellt.

Jaro-Winkler-Ähnlichkeit Die Jaro-Winkler-Ähnlichkeit (Winkler, 1990; Cohen et al., 2003) baut auf der Jaro-Distanz (Jaro, 1989) auf. Die Jaro-Distanz d_j ist ein auf $[0, 1]$ normiertes Distanzmaß, jedoch keine Metrik (siehe Definition A.2), die gegeben ist durch

$$d_j = \begin{cases} 1, & \text{für } m = 0, \\ 1 - \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{sonst.} \end{cases}$$

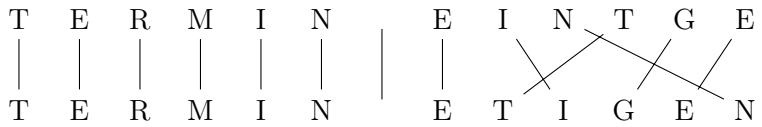


Abbildung 2.5: Beispiel für Transpositionen bei der Jaro-Distanz.

Der Parameter m zählt die übereinstimmenden Zeichen in den beiden Zeichenketten s_1 und s_2, wobei zwei Zeichen übereinstimmen, wenn sie gleich sind und ihre Positionen im jeweiligen String nicht weiter als $\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} \right\rfloor - 1$ voneinander entfernt sind. Das heißt, dass übereinstimmende Zeichen nicht notwendigerweise an den gleichen Positionen stehen müssen, sondern abhängig von der Gesamtlänge des längeren Strings nur nah genug. Der Parameter t zählt die Anzahl an Transpositionen, d. h. Vertauschungen, als die Anzahl an übereinstimmenden Zeichen, die auf falschen Positionen stehen, geteilt durch zwei mit abrunden auf die nächst kleinere ganze Zahl. Eine Transposition muss dann vorgenommen werden, wenn Zeichen zwar übereinstimmen, d. h. ihre Positionen nah genug beieinander liegen, ihre Reihenfolgen in den beiden Strings jedoch vertauscht sind. Es ist immer $t \leq m$, außerdem gilt $d_j \in [0, 1]$.

Beispiel 2.2. Für die beiden Zeichenketten s_1 =„Termin eintragen“ und s_2 =„Termin bestätigen“ ist $16 = |s_1| < |s_2| = 17$ und somit ist der Abstand, den gleiche Zeichen maximal haben dürfen, um als übereinstimmend zu gelten:

$$\left\lfloor \frac{\max(16, 17)}{2} \right\rfloor - 1 = \left\lfloor \frac{17}{2} \right\rfloor - 1 = 7$$

In Tabelle 2.1 ist notiert, für welche Zeichen ein gleiches Zeichen innerhalb des Toleranzbereiches mit Abstand 7 gefunden wird. Jedes Zeichen kann hierbei nur maximal einmal ein passendes Gegenstück im anderen String haben. Es wird über den kürzeren String iteriert und von der unteren Grenze des Toleranzbereichs die jeweils erste, neue Entsprechung gewertet (◯). Für dieses Beispiel ist $m = 13$, wobei für s_1 die übereinstimmenden Zeichen T-E-R-M-I-N-_-E-I-N-T-G-E und für s_2 die übereinstimmenden Zeichen T-E-R-M-I-N-_-E-T-I-G-E-N sind. Hierbei sind, wie Abbildung 2.5 zeigt, nicht alle übereinstimmenden Zeichen auf den richtigen Positionen. Fünf Zeichen passen nicht, was zu $t = \lfloor 5/2 \rfloor = 2$ Transpositionen führt. Damit ergibt sich, da $m \neq 0$, folgende Jaro-Distanz:

$$d_j = 1 - \frac{1}{3}\left(\frac{13}{16} + \frac{13}{17} + \frac{11}{13}\right) \approx 0,19$$

Es sei angemerkt, dass der Algorithmus, sich entsprechende Zeichen zu finden, nicht optimal ist. Der Distanzwert wäre in diesem Beispiel geringer, würde das letzte N aus „Termin eintragen“, und nicht das N an zehnter Stelle, auf das letzte N aus „Termin bestätigen“ gematcht. Denn dann wäre die Anzahl der Transpositionen, die positiv mit der Größe der Distanz korreliert ist, geringer. Als Ähnlichkeit der beiden Strings kann ein Wert von 81% mittels $1 - d_j$ angegeben werden.

Aus der Jaro-Distanz wird die Jaro-Winkler-Distanz (Winkler, 1990) abgeleitet, die diejenigen Strings begünstigt, die ein gleiches Präfix, also eine gleiche Vorsilbe, haben. Die Länge des gemeinsamen Präfixes wird mit ℓ bezeichnet, wobei für ℓ aus Erfahrung festgelegt wird, dass es nie größer als vier sein kann. Über einen festen Skalierungsparameter p ($p \leq \frac{1}{4}$, damit

Tabelle 2.1: Beispiel für Jaro-Distanz (∘ = Treffer, ∼ = schon benutzt, × = falsch; alle unausgefüllten Felder außerhalb des Toleranzbereichs).

	T	E	R	M	I	N		B	E	S	T	Ä	T	I	G	E	N
T	◯	×	×	×	×	×	×	×									
E	×	◯	×	×	×	×	×	×	∼								
R	×	×	◯	×	×	×	×	×	×	×							
M	×	×	×	◯	×	×	×	×	×	×	×						
I	×	×	×	×	◯	×	×	×	×	×	×	×					
N	×	×	×	×	×	◯	×	×	×	×	×	×	×				
	×	×	×	×	×	×	◯	×	×	×	×	×	×	×			
E	×	∼	×	×	×	×	×	×	◯	×	×	×	×	×	×		
I		×	×	×	∼	×	×	×	×	×	×	×	×	◯	×	×	
N			×	×	×	∼	×	×	×	×	×	×	×	×	×	×	◯
T				×	×	×	×	×	×	×	◯	×	∼	×	×	×	×
R					×	×	×	×	×	×	×	×	×	×	×	×	×
A						×	×	×	×	×	×	×	×	×	×	×	×
G							×	×	×	×	×	×	×	×	◯	×	×
E								×	∼	×	×	×	×	×	×	◯	×
N									×	×	×	×	×	×	×	×	∼

$\ell p \leq 1$ für alle $\ell \leq 4$) wird bestimmt, wie stark gleiche Präfixe in die Jaro-Distanz einfließen. Die Jaro-Winkler-Distanz d_w ist dann definiert als

$$d_w = d_j + (\ell p(1 - d_j)) \in [0, 1].$$

Die Jaro-Winkler-Distanz ist damit größer gleich der Jaro-Distanz. Strings mit gleichen Präfixen werden bei Jaro-Winkler als ähnlicher eingestuft als Strings mit verschiedenen Präfixen im Vergleich zur Einstufung durch die Jaro-Ähnlichkeit. Die Jaro-Winkler-Ähnlichkeit sim_{jw} lässt sich einfach über

$$sim_{jw} = 1 - d_w \in [0, 1]$$

angeben. Der Einsatz von Stemming im Zusammenhang mit der Jaro-Winkler-Ähnlichkeit ist sorgfältig zu prüfen, da sich Präfixe durch Stemming ändern können.

Jaccard-Ähnlichkeit/Tanimoto-Ähnlichkeit Bei der Jaccard-Ähnlichkeit (Cohen et al., 2003) werden die zu vergleichenden Strings als Mengen von Wörtern betrachtet und der Anteil gleicher Wörter errechnet. Der Ausdruck $w(s_i)$ bezeichne die Menge der Wörter in String s_i, $i = 1, 2$.

$$sim_{jac} = \frac{|w(s_1) \cap w(s_2)|}{|w(s_1) \cup w(s_2)|} \in [0, 1]$$

N-Gramme und Dice-Koeffizient Der Dice-Koeffizient (Fluri et al., 2007) ist ein Ähnlichkeitsmaß für Zeichenketten, wobei er, ähnlich wie der Jaccard-Koeffizient, die Anzahl der gleichen Substrings, die in den zu vergleichenden Zeichenketten vorkommen, in Relation setzt zur Anzahl aller Substrings. Als Substrings werden hierbei N-Gramme, meist Bi- oder Trigramme verwendet. Ein N-Gramm ist hier ein Textfragment aus N aufeinanderfolgenden Zeichen. So wird beispielsweise das Wort „Haus“ in die 2-Gramme/Bigramme „Ha“, „au“ und

„us“ zerlegt. Es bezeichne $ng_N(s_1)$ die Menge der N-Gramme von String s_1 und $ng_N(s_2)$ die Menge der N-Gramme von String s_2. Der Dice-Koeffizient ist dann

$$sim_{diceN}(s_1, s_2) = \frac{2|ng_N(s_1) \cap ng_N(s_2)|}{|ng_N(s_1)| + |ng_N(s_2)|} \in [0, 1].$$

Alternativ wäre hier auch ein Jaccard-Ansatz auf N-Grammen möglich:

$$sim_{jacN} = \frac{|ng_N(s_1) \cap ng_N(s_2)|}{|ng_N(s_1) \cup ng_N(s_2)|}$$

Anmerkung Allgemein gilt für die Mengen A und B mit $jac(A, B) = \frac{|A \cap B|}{|A \cup B|}$ und $dice(A, B) = \frac{2|A \cap B|}{|A|+|B|}$:

- Falls $A \cap B = \emptyset$, dann ist $jac(A, B) = 0 = dice(A, B)$.
- Falls $A \cap B \neq \emptyset$ und somit $jac(A, B) \neq 0 \neq dice(A, B)$, dann ist wegen $|A \cap B| \leq |A \cup B|$

$$\begin{aligned} \frac{2}{dice(A, B)} &= \frac{|A| + |B|}{|A \cap B|} \\ &= \frac{|A \cap B| + |A \cup B|}{|A \cap B|} \\ &\leq \frac{2|A \cup B|}{|A \cap B|} \\ &= \frac{2}{jac(A, B)} \\ \Leftrightarrow jac(A, B) &\leq dice(A, B). \end{aligned}$$

Der Jaccard-Koeffizient ist also im Allgemeinen strenger als der Dice-Koeffizient. Lediglich in den Randfällen ($A \cap B = \emptyset$ und $A = B$) liefern sie die gleichen Werte (0 bzw. 1).

Weitere Stringmetriken sind beispielsweise die Hamming-Distanz (Hamming, 1950), die L1-Distanz/Manhattan-Metrik, der Tversky-Index (Tversky, 1977), die Hellinger-Distanz (Hellinger, 1909), der Overlap-Koeffizient (Bradley, 2004) oder die (Kendall-) Tau-Metrik (Kendall, 1938, 1948). Alle diese Metriken ließen sich auf M:N-Abgleiche anpassen, doch die tatsächliche Aussagekraft dieser rein syntaktischen Betrachtungen ist fraglich. In Abschnitt 4.1 wird deshalb, und weil noch weitere Perspektiven außer der Aktivitätenbeschreibung berücksichtigt werden, ein relativ einfaches syntaktisches Ähnlichkeitsmaß betrachtet.

Wie auch Paice (1994) zeigt, sind die Unterschiede der aufgeführten Stringähnlichkeitsmethoden im Allgemeinen eher gering. Die Jaro-Ähnlichkeit erfordert keine externen Parameter, die Jaro-Winkler-Ähnlichkeit benötigt die Angabe der Präfixlänge, die von der verwendeten Sprache abhängig gewählt werden kann. Die Jaccard-Ähnlichkeit ist dadurch, dass die Aktivitätenbeschreibungen zunächst in Mengen von Wörtern umgewandelt werden, etwas aufwändiger. Gerade bei standardisiertem Vokabular ist diese Methode jedoch gut anzuwenden (Paice, 1994). Rechtschreibfehler in Wörtern können die berechneten Ähnlichkeiten jedoch erheblich mindern, d. h. die Wahl der Ähnlichkeitsmethode hängt auch von der erwarteten Güte der Aktiviätenbeschreibungen ab. Der Dice-Koeffizient stellt einen Mittelweg zwischen zeichen- und wörterbasierten Methoden dar. Die Länge der N-Gramme kann beispielsweise wie die Länge der Präfixe der Jaro-Winkler-Ähnlichkeit gewählt werden. Rechtschreibfehler fallen

hier wie bei den zeichenbasierten Methoden weniger ins Gewicht. Die gezeigten Methoden können ebenfalls zuvor mit Stemmingalgorithmen vorverarbeitet werden.

2.2.2.4 Semantische/Lexikalische/Linguistische Ähnlichkeit

Dijkman et al. (2011) und Ehrig et al. (2007) schlagen unter anderem die Verwendung von Wörterbüchern vor, um die Bedeutung von Beschriftungen bei der Ähnlichkeitsberechnung ebenfalls zu berücksichtigen. Ein solches Ähnlichkeitsmaß wird von Ehrig et al. (2007) vorgestellt und bezieht Synonyme, Generalisierungen und Spezialisierungen, die bei rein syntaktischer Labelbehandlung nicht erkannt werden, in den Ähnlichkeitsabgleich mit ein. Prozessmodelle, die in Form von Petri-Netzen vorliegen, werden von Ehrig et al. (2007) in eine Ontologie-basierte Form gebracht, wobei hier konkret die Web Ontology Language (OWL) verwendet wird (Antoniou und van Harmelen, 2004; W3C OWL Working Group, 2012). Die so transformierten Modelle werden semantische Prozessmodelle genannt.

Allgemein ist eine Ontologie eine formal geordnete Darstellung von Begrifflichkeiten, Konzepte genannt, und deren gegenseitigen Beziehungen in einem bestimmten Kontext (Hesse, 2002). Konzepte können in einer Subsumptionshierarchie angeordnet sein und werden durch bestimmte Eigenschaften beschrieben, die wiederum hierarchisch angeordnet sein können. Koschmider und Oberweis (2005) beschreiben den Vorgang, Petrinetze in semantische Prozessmodelle zu überführen.

Die Ähnlichkeit von semantischen Prozessmodellen wird von Ehrig et al. (2007) aus drei Komponenten berechnet, wobei eine davon die syntaktische Ähnlichkeit $sim_{sed'}$ gemäß Gleichung (2.4) darstellt. Die zweite Komponente, die linguistische Ähnlichkeit, verwendet Synonymbeziehungen, die im Wörterbuch WordNet (Miller, 1995) definiert sind. Die dritte verwendet die Struktur der semantischen Prozessmodelle. Die linguistische Ähnlichkeit wird nur für die Wörter der beiden zu vergleichenden Zeichenketten berechnet, deren syntaktische Ähnlichkeit keinen Wert von 1 zurückgibt. Das heißt, bei einem Vergleich der Strings „versende Bestätigung“ und „versende Bescheinigung“ wird die linguistische Ähnlichkeit nur von den Wörten „Bestätigung“ und „Bescheinigung“ ermittelt, da $sim_{sed'}(\text{versende}, \text{versende}) = 1$.

Beim Berechnen der linguistischen Ähnlichkeit gehen Ehrig et al. (2007) wie folgt vor: Die Funktion $\eta(c)$ gibt die Menge aller in WordNet gelisteten Synonyme von c aus, wobei c eine Konzeptinstanz, also eine Zeichenkette bestehend aus Wörtern, ist. Die Indikatorfunktion $\mathbb{1}_{\eta(c_1)\cap\eta(c_2)\neq\emptyset}$ gibt an, ob die beiden Konzeptinstanzen c_1 und c_2 gleiche Synonyme haben. Mit der Formel

$$sim_{ling}(c_1, c_2) = \frac{\mathbb{1}_{\eta(c_1)\cap\eta(c_2)\neq\emptyset}}{\max(|\eta(c_1)|, |\eta(c_2)|)} \in [0, 1]$$

wird dann die linguistische Ähnlichkeit berechnet. Diese Ähnlichkeit ist maximal, wenn c_1 das einzige Synonym von c_2 ist und umgekehrt. Die linguistische Ähnlichkeit nimmt also ab, je mehr alternative Formulierungen für ein Wort existieren. Von Ehrig et al. (2007) wird nicht genau spezifiziert, wie die linguistische Ähnlichkeit von Zeichenketten bestimmt wird, bei denen jeweils mehr als ein Wort verglichen werden muss, also wie sie bei den Zeichenketten „versende Bestätigung“ und „verschicke Bescheinigung“ berechnet wird, da die syntaktische Ähnlichkeit für keine Kombination an Wörtern den Wert 1 annimmt.

Die dritte Komponente bei der Messung der Ähnlichkeit von semantischen Prozessmodellen wird von Ehrig et al. (2007) als strukturelle Ähnlichkeit bezeichnet, jedoch ist der Begriff „Struktur“ in diesem Zusammenhang nicht zu verwechseln mit der Struktur von Prozessmodellen, wenn diese gemäß Definition 1.1 gegeben sind. Die Struktur, die von Ehrig

et al. (2007) genannt wird, bezieht sich auf die Struktur der ontologiebasierten Darstellung der Prozessmodelle, also die hierarchische Einordnung der Konzepte. Zu jedem Konzept wird ein Kontext, das ist ein Tupel aus Mengen von Attributen, Attributtypen usw., festgelegt. Im Ausschnitt in Abbildung 2.6 ist die Menge an Attributen des Konzepts „Anfrage“ beispielsweise {Name, Datum, Uhrzeit}. Die strukturelle Ähnlichkeit zweier Konzepte errechnet sich dann aus der syntaktischen und linguistischen Ähnlichkeit der Kontextelemente, die über eine bestimmte Gewichtung miteinander verrechnet werden. Homonyme können auf diese Weise entdeckt werden. Zuletzt werden die drei genannten Komponenten, syntaktische, linguistische und strukturelle Ähnlichkeit, ähnlich wie beim Vorgehen des Ähnlichkeitsabgleichs aus Abschnitt 2.2.1, mit einer Gewichtung zu einem Ähnlichkeitswert zusammengeführt.

Die Berechnung der strukturellen Ähnlichkeit ist so konstruiert, dass sie nicht symmetrisch ist, was sich auf den kombinierten Ähnlichkeitswert überträgt, sodass dieser nicht als Ähnlichkeitsmaß, wie in Definition 2.1 gefordert, Anwendung finden kann. Aus diesem Grund ist dieses Ähnlichkeitsmaß, selbst wenn es auf M:N-Abbildungen angepasst würde, für den Fortlauf der Arbeit weniger geeignet.

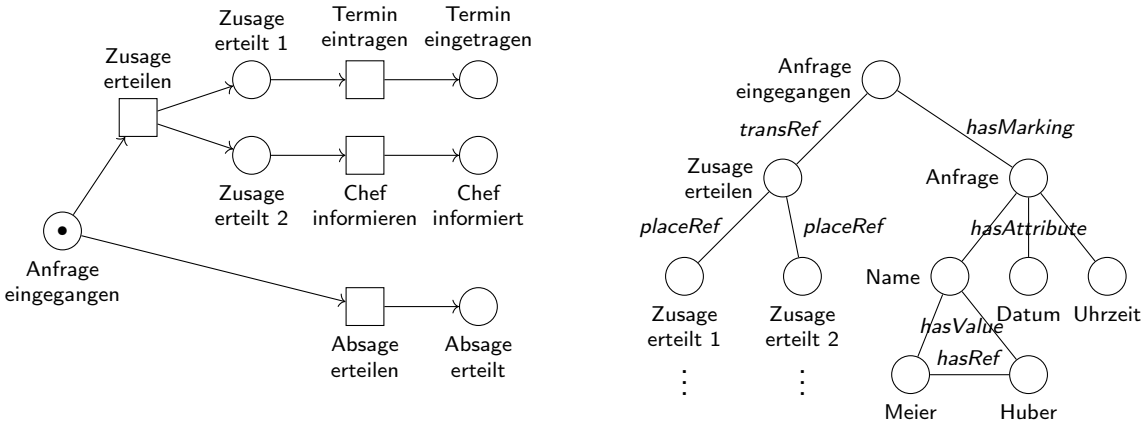


Abbildung 2.6: Beispiel für ein Petrinetz (links) und ein daraus abgeleitetes semantisches Prozessmodell (rechts, Ausschnitt)

2.2.2.5 Bag of Words-Ähnlichkeit

Die Bag of Words-Ähnlichkeit wird von Klinkmüller et al. (2013) als Möglichkeit vorgeschlagen, Prozessmodelle anhand ihrer Aktivitätsbeschreibungen abzugleichen. Das Prozessmodell wird auf die Menge aller Aktivitätsbeschreibungen reduziert, die wiederum als Menge an Wörtern behandelt werden. Diese Aufgabe übernimmt eine Funktion z, die die Menge aller Beschriftungen $\mathcal{L}$ auf die Potenzmenge aller Wörter, die in der Menge $\mathcal{W}$ der Wörter zusammengefasst sind, abbildet, jedes Label also in seine Wörter zerlegt: $z : \mathcal{L} \rightarrow \mathcal{P}(\mathcal{W})$. Anschließend werden noch Stoppwörter (siehe Abschnitt 2.2.2.2) aus der so entstandenen Wörtermenge entfernt. Das Bag of Words-Modell ist die Zerlegung eines Textes in 1-Gramme auf Wörterbasis, nicht auf Zeichenbasis, und kann als Alternative zur Jaccard-Ähnlichkeit angesehen werden. Die Struktur innerhalb der Beschreibungen wird nicht beachtet, was von Klinkmüller et al. (2013) mit der Kürze der Label und der daraus resultierenden, geringen Aussagekraft der Labelstruktur begründet wird.

Das Bag of Words-Ähnlichkeitsmaß gleicht die Menge an Wörtern einer Aktivität mit

der Menge an Wörtern einer anderen Aktivität ab. Genauer: Es soll die Ähnlichkeit von Aktivität a_1 aus Prozessmodell $G_1 = (N_1, E_1, \lambda)$ mit $a_1 \in A_1 \subseteq N_1$ und Aktivität a_2 aus Prozessmodell $G_2 = (N_2, E_2, \lambda)$ mit $a_2 \in A_2 \subseteq N_2$ bestimmt werden. Die Variable $w_{l,k}$, $k = 1, \ldots, W_l$, bezeichnet ein Wort aus $z(\lambda(a_l))$ und W_l die Mächtigkeit der entsprechenden Menge an Wörtern, also $W_l = |z(\lambda(a_l))|$, $l = 1, 2$. Die Funktion sim ist eine Funktion, die zwei Wörtern einen Ähnlichkeitswert aus dem Intervall $[0, 1]$ zuordnet. Für sim kann beispielsweise sim_{sed} oder sim_{ling} verwendet werden. Das Bag of Words-Ähnlichkeitsmaß ist dann:

$$sim_{bow}(a_1, a_2) = \frac{\sum_{i=1}^{W_1} \max_{j=1,\ldots,W_2} (sim(w_{1,i}, w_{2,j})) + \sum_{j=1}^{W_2} \max_{i=1,\ldots,W_1} (sim(w_{1,i}, w_{2,j}))}{W_1 + W_2} \in [0, 1]$$

Das heißt, für jedes Wort der Beschreibung von a_1 wird das ähnlichste Wort, bezogen auf sim, aus a_2 gesucht. Anders herum wird genau so verfahren, um die Symmetrie des Maßes zu gewährleisten. Anschließend werden diese Ähnlichkeitswerte aufsummiert und durch deren Anzahl geteilt; es wird also über die Ähnlichkeitswerte der einzelnen Wortpaare gemittelt. Bei diesem Verfahren geht jedes Wort mindestens einmal in die Berechnung ein (Iteration beim Summenzeichen), wobei es durchaus vorkommen kann, dass die maximale Ähnlichkeit mehrmals dasselbe Wort im anderen Modell adressiert. Auf diese Weise ist es tendenziell möglich, dass wichtige Wörter, also solche, die das Thema einer Beschreibung darstellen, stärker in die Berechnung eingehen, als solche, die überhaupt keine Entsprechung haben. Letztere senken den Ähnlichkeitswert zwar, aber nur geringfügig. Dadurch, dass immer die beste Entsprechung gesucht wird, ist generell ein recht hoher Wert für die Bag of Words-Ähnlichkeit zu erwarten. Problematisch ist dieses Verfahren hingegen, wenn sich der Sinn einer Aussage beispielsweise durch Verwendung des Wortes „nicht“ komplett herumdreht. Unterscheiden sich zwei Aktivitätenbeschreibungen nur durch dieses Wort, so wird deren Ähnlichkeitswert doch sehr nahe bei 1 liegen, d. h., die Bag of Words-Methode versagt in diesem Fall. Diesem Problem könnte beispielsweise dadurch begegnet werden, dass Beschriftungen stets positiv formuliert werden. Da bei stark unterschiedlich langen Labels viele möglicherweise unwichtige Wörter des längeren Labels in gleichem Maße berücksichtigt werden wie die wichtigen und die Ähnlichkeit somit stark reduzieren, ist die Bag of Words-Ähnlichkeit für M:N-Abbildungen und somit für die weitere Arbeit in der gezeigten Form nicht gut geeignet. Klinkmüller et al. (2013) schlagen jedoch selbst eine dahingehende Verbesserung der Methode vor, die in Abschnitt 2.2.2.6 gezeigt wird und die zu lange Labels auf die wichtigen Wörter reduziert, wobei „wichtig“ unterschiedlich definiert werden kann.

2.2.2.6 Bag of Words-Ähnlichkeit mit Label Pruning

Das in diesem Abschnitt vorgestellte Ähnlichkeitsmaß wird ebenfalls von Klinkmüller et al. (2013) definiert und baut auf der Bag of Words-Ähnlichkeit auf, versucht aber Labels mit unterschiedlich großer Wörterzahl besser zu beurteilen als die reine Bag of Words-Ähnlichkeit. Insbesondere Labels mit unterschiedlicher Beschreibungsgenauigkeit derselben Aufgabe sollen dadurch eine höhere Ähnlichkeit erhalten. Betrachte z. B. die beien Label „Termin eintragen“ und „Termin in den Kalender eintragen“. In der einen Bag of Words-Ähnlichkeit würden auch die drei Wörter „in“, „den“ und „Kalender“ bei der Ähnlichkeitsberechnung berücksichtigt werden und den Wert stark senken, da im anderen String keine guten Wortentsprechungen vorhanden sind. Werden diese drei Wörter jedoch als vergleichsweise unwichtig eingestuft und

bei der Berechnung der Ähnlichkeit ausgeklammert, erhöhnt sich die Ähnlichkeit der beiden Label erheblich. Dieses Einschränken der Wörterzahl wird „label pruning“ genannt und durch die nicht symmetrische Funktion $p : \mathcal{P}(\mathcal{W}) \times \mathcal{P}(\mathcal{W}) \to \mathcal{P}(\mathcal{W})$ realisiert, die durch folgende Fallunterscheidung gegeben ist:

$$p(z(\lambda(a_1)), z(\lambda(a_2))) = \begin{cases} z(\lambda(a_1)), & \text{falls } W_1 \leq W_2 \\ p(z(\lambda(a_1))) \text{ mit } p(z(\lambda(a_1))) \subsetneq z(\lambda(a_1)) & \\ \qquad \wedge\ |p(z(\lambda(a_1)))| = W_2, & \text{falls } W_1 > W_2 \end{cases}$$

Ohne Beschränkung der Allgemeinheit habe im Folgenden Aktivität a_1 eine Beschreibung mit mehr Wörtern als Aktivität a_2, also $W_1 = |z(\lambda(a_1))| > |z(\lambda(a_2))| = W_2$. Beim Berechnen der Ähnlichkeit von a_1 und a_2 werden von der Beschreibung von a_1 nur W_2-viele Wörter berücksichtigt. Zur Auswahl dieser Wörter, also der konkreten Ausgestaltung von Funktion p, gehen Klinkmüller et al. (2013) auf drei Möglichkeiten ein.

- Bei der ersten Variante wird die Ähnlichkeit aller Wortpaare berechnet, was es ermöglicht, für jedes Wort aus $z(\lambda(a_1))$ den höchsten Ähnlichkeitswert zu ermitteln. Die Wörter aus $z(\lambda(a_1))$ werden anhand dieses maximalen Ähnlichkeitswerts sortiert und die W_2-vielen Wörter mit den größten Ähnlichkeitswerten aus dem Label von a_1 ausgewählt. Die Funktion p heißt dann p_{max}. Die Wörter, deren beste Entsprechung im anderen Modell zu gering ist, werden bei der Berechnung des Ähnlichkeitswerts der kompletten Beschreibung, also bei der Mittelung über die einzelnen Wörterähnlichkeiten, gar nicht berücksichtigt.

Für die anderen beiden Varianten wird die relative Häufigkeit von Wörtern mit einbezogen, genannt Dokumenthäufigkeit (*document frequency*), ein Begriff, der vor allem im Bereich des Information Retrieval eine Rolle spielt. Ein Dokument ist in diesem Fall die Menge der Wörter einer Aktivitätenbeschreibung. Es wird gezählt, in wie vielen Dokumenten ein Term t, also ein Wort, auftaucht. Die beiden von Klinkmüller et al. (2013) genannten Möglichkeiten unterscheiden sich darin, welche Dokumente, d. h. welche Aktivitäten, bei der Bestimmung der Häufigkeit berücksichtigt werden.

- In dieser zweiten Variante werden als Dokumente alle Aktivitäten aller Prozessmodelle im Repositorium, die jeweils paarweise miteinander verglichen werden, zugelassen. Besteht das Repositorium aus den Prozessmodellen $G_1, \ldots, G_n$ mit den Aktivitäten $A_1, \ldots, A_n$ und bezeichnet $\mathcal{D}$ die Menge aller Dokumente, dann ist $|\mathcal{D}| = |A_1| + \ldots + |A_n|$. Die relative Häufigkeit eines Terms t ist dann $\frac{h_t}{|\mathcal{D}|}$, wobei h_t die Anzahl der Dokumente aus $\mathcal{D}$, in denen t auftaucht, ist. Die Funktion p heißt dann p_{coll}.
- In der dritten Variante werden als Dokumente nur die Aktivitäten der beiden beteiligten Prozessmodelle zugelassen. Wenn $\mathcal{D}$ die Menge aller Dokumente bezeichnet, dann ist $|\mathcal{D}| = |A_1| + |A_2|$. Die Funktion p heißt dann p_{2p}.

Die Bag of Words-Ähnlichkeit mit Label Pruning ist damit wie folgt gegeben, wobei $pr_1 = p(z(\lambda(a_1)), z(\lambda(a_2)))$ und $pr_2 = p(z(\lambda(a_2)), z(\lambda(a_1)))$:

$$sim_{lp}(a_1, a_2) = \frac{\sum_{i=1}^{W_1} \max_{j=1,\ldots,W_2} (sim(pr_{1,i}, pr_{2,j})) + \sum_{j=1}^{W_2} \max_{i=1,\ldots,W_1} (sim(pr_{1,i}, pr_{2,j}))}{2 \cdot \min(W_1, W_2)} \in [0, 1]$$

Die Berücksichtigung von unterschiedlichen Labellängen wird von Klinkmüller et al. (2013) als Stärke dieses Ansatzes aufgeführt. Deswegen scheint diese Art der Ähnlichkeitsberechnung von Aktivitätenbeschreibungen sehr gut geeignet auch für M:N-Abbildungen zu sein, da dort, wie in Abschnitt 4.1 erläutert, eine stark differierende Wörteranzahl in den Beschreibungen der Aktivitäten zu erwarten ist. Die Bag of Words-Ähnlichkeit mit Label Pruning gleicht dadurch, dass bei einem Überschuss an Wörtern die mit den geringsten Einzelänlichkeitswerten weggelassen werden, schlechte Ähnlichkeitswerte aus, die wegen der Mittelung stark ausreißerempfindlich sind. Ein Nachteil dieses Verfahrens ist, dass möglicherweise sinnentscheidende Wörter so gar nicht mehr bei der Ähnlichkeitsberechnung berücksichtigt werden, während diese bei der Variante ohne Label Pruning zumindest ein wenig Einfluss auf den Ähnlichkeitswert haben. Unter Verwendung von p_{max} haben beispielsweise die beiden Strings „Ich liebe dich“ und „Ich liebe dich nicht“ eine Ähnlichkeit von 1. Eine kurze Diskussion erfolgt dann ebenfalls in Abschnitt 4.1.

2.2.2.7 Mittelung über Ähnlichkeit von Wortpaaren

Auch von La Rosa et al. (2010) wird eine linguistische Ähnlichkeit, sim_{wp}, vorgeschlagen: Zwei Aktivitätsbeschreibungen werden miteinander verglichen, indem jedes Wort der einen Beschreibung mit jedem Wort der anderen Beschreibung verglichen wird. Für die Ähnlichkeit der Wortpaare wird, nach Anwendung von Stemming-Methoden, das Maximum aus String-Editierabstand und linguistischer Ähnlichkeit, z. B. Test auf Synonyme, gewählt. Die Ähnlichkeit zweier Beschreibungen ist dann der Mittelwert über die Ähnlichkeiten der einzelnen Wortpaare. Dadurch, dass bei dieser Ähnlichkeitsberechnung jedes Wort mit jedem verglichen wird und nicht, wie bei der Bag of Words-Ähnlichkeit, nur mit der jeweils besten Entsprechung, liefert dieses Verfahren selbst bei gleichen Beschreibungen nicht notwendigerweise einen Ähnlichkeitswert von 1. Die Ähnlichkeit von „Chef informieren“ mit sich selbst ist, wenn der String-Editierabstand für die Ähnlichkeit der einzelnen Wörter angewendet wird, mit $c =$ „Chef“ und $i =$ „informieren“: $sim_{wp} = {}^1\!/_4 \cdot (sim_{sed}(c,c) + sim_{sed}(c,i) + sim_{sed}(i,c) + sim_{sed}(i,i)) = {}^1\!/_4 \cdot (2 + {}^2\!/_{11}) = {}^{24}\!/_{44} \approx 0,55$. Dieses Verfahren erfüllt also nicht die Bedingung aus Definition 2.1 für ein Ähnlichkeitsmaß und ist in der gegebenen Form für den weiteren Verlauf der Arbeit nicht relevant.

Welche der aufgeführten Ähnlichkeitsmethoden für Aktivitätsbeschreibungen verwendet werden sollte, hängt von den zu vergleichenden Prozessmodellen, insbesondere deren Labelformulierungen, ab. Die semantische Ähnlichkeit kann nur dann angewendet werden, wenn ein geeignetes Wörterbuch oder eine Ontologie zur Verfügung stehen. Auch dann ist der Aufwand, was Zeit und Komplexität der Implementierung angeht, relativ hoch. Bei frei formulierten Beschreibungen, die wenige bis gar keine Fehler enthalten, sind für diese Methode jedoch die besten Ergebnisse zu erwarten, unter anderem weil Homonyme beachtet werden können. Wörterbasierte Abgleichsmethoden, zum Beispiel die Bag of Words-Ähnlichkeit, lassen bei einem standardisierten Vokabular der Beschreibungen ebenfalls gute Ergebnisse erwarten, wobei der Aufwand im Vergleich zu semantischen Methoden geringer ist. Syntaktische Ähnlichkeitsmaße auf Zeichenebene sind die tendenziell am einfachsten zu berechnenden, außerdem fallen kleinere Fehler in den Beschreibungen relativ wenig ins Gewicht. Eine Verwandtschaft von Wörtern oder eine dahinterliegende Bedeutung werden jedoch komplett ignoriert. In Zusammenhang mit Stemming-Methoden oder bei standardisiertem Vokabular können aber auch hier gute Ergebnisse erzielt werden.

2.2.3 Strukturbasierte Ansätze zum Ähnlichkeitsabgleich

Durch das Einbeziehen der abgebildeten Kanten ist auch im vierstufigen Ansatz, wie er in Abschnitt 2.2.1 beschrieben ist, schon ein Teil der Struktur von Prozessmodellen mit berücksichtigt. Fasst man Prozessmodelle jedoch als Graphen mit gerichteten Kanten auf, so ergeben sich noch weitere Möglichkeiten, Prozessmodelle abzugleichen (Minor et al., 2007). Die strukturbasierten Ansätze, die im Folgenden präsentiert werden, setzen eine Abbildung zwischen den beiden zu vergleichenden Prozessmodellen voraus. Ist von gleichen Knoten in unterschiedlichen Knotenmengen die Rede, so sind immer die durch die Abbildung festgelegten Korrespondenzen gemeint, d. h., die Abbildung legt fest, welche Knoten gleich sind. Einige Aspekte der strukturbasierten Ähnlichkeitsmaße, wie die merkmalsbasierte Betrachtung von Aktivitäten oder das Einbeziehen der Datenflussperspektive, finden sich in Kapitel 4 wieder.

Die strukturbasierten Abgleichsansätze umfassen Methoden, die nur die Knoten der Modelle betrachten (Abschnitte 2.2.3.1, 2.2.3.2), nur die Kanten (Abschnitt 2.2.3.3), aber auch Knoten und Kanten (Abschnitt 2.2.3.4). In den Abschnitten 2.2.3.5 und 2.2.3.6 werden Methoden, die Vorgänger- und Nachfolgerbeziehungen berücksichtigen, vorgestellt, also über das bloße Vorhandensein von Modellelementen hinausgehen. Die Ähnlichkeitsdefinitionen aus den Abschnitten 2.2.3.7, 2.2.3.8 und 2.2.3.9 beziehen detailliertere Eigenschaften von Modellen inklusive unterschiedlicher Ausführungssemantik ein, wobei die Bedeutung der Semantik nicht näher verwendet wird. In Abschnitt 2.2.3.10 werden Prozessmodelle schließlich als Graphen als Ganzes betrachtet, wobei mit der Definition des Editierabstands von Graphen Parallelen zum Anteil abgebildeter Knoten und Kanten aus dem vierstufigen Abgleichsansatz (Abschnitt 2.2.1) gezogen werden.

2.2.3.1 Einfache, semantische Ähnlichkeit

Der Ansatz von Akkiraju und Ivan (2010) zur Bestimmung der Ähnlichkeit zweier Prozessmodelle identifiziert Aktivitäten über ihre Beschreibungen und setzt somit ursprünglich die Identitätsabbildung voraus. Jedoch kann die Berechnung dieses Ähnlichkeitsmaßes auch für jede andere Abbildung gemäß Definition 2.2 erfolgen. Der Ähnlichkeitswert errechnet sich über den Anteil gleichlautender Aktivitätsbeschreibungen bzw. den Anteil abgebildeter Aktivitäten:

$$fsim_{nodes}(G_1, G_2) = \frac{2 \cdot |N_1 \cap N_2|}{|N_1| + |N_2|} \in [0, 1],$$

wobei die Bezeichnungen wie in Definition 2.2 gewählt sind. Die Berechnung dieser Ähnlichkeit entspricht der des Dice-Koeffizienten aus Abschnitt 2.2.2.3 mit dem Unterschied, dass hier nicht aus zwei Mengen an N-Grammen ein Ähnlichkeitswert gebildet wird sondern aus zwei Mengen an Knoten. Eine Abbildung ergibt sich dann indirekt, indem die gleich benannten Aktivitäten abgeglichen werden. Diese Möglichkeit der Ähnlichkeitsbestimmung ist, wenn sie wie von Akkiraju und Ivan (2010) mit der Identitätsabbildung verwendet wird, nur dann sinnvoll, wenn ein standardisiertes Vokabular bei der Modellierung zur Verfügung steht. Duplikate werden damit gefunden, ebenso Varianten von Prozessen. Die Struktur von Prozessmodellen, also zum Beispiel die Reihenfolge der Aktivitäten, wird allerdings gänzlich außer Acht gelassen (Becker und Laue, 2012). Unter diesem Gesichtspunkt ist der Ansatz für den Fortgang dieser Arbeit als unzureichend einzustufen, denn z. B. aus Compliancesicht kann die Reihenfolge von Aktivitäten eine wichtige Rolle spielen.

2.2.3.2 δ-Vergleichbarkeit

Von Bae et al. (2006b) wird ein Ansatz ähnlich zu dem aus Abschnitt 2.2.3.1 angesprochen, allerdings bezogen auf Abhängigkeitsgraphen (siehe auch Abschnitt 2.2.3.8), die hier jedoch auch als Prozessmodelle gemäß Definition 1.1 aufgefasst werden können. Zwei Abhängigkeitsgraphen erhalten den Ähnlichkeitswert 1, wenn ihre Knotenmengen die gleichen sind, 0, wenn ihre Knotenmengen komplett verschieden sind, und einen Wert $k \in [0, 1]$, wenn sich ihre Knotenmengen teilweise überschneiden; k ist hierbei der Anteil der gemeinsamen Knoten im Vergleich zur Vereinigung aller Knoten: $|DN_1 \cap DN_2| / |DN_1 \cup DN_2|$. Zwei Abhängigkeitsgraphen sind sich δ-ähnlich, wenn ihr Ähnlichkeitswert k den Grenzwert δ überschreitet, $\delta \in (0, 1]$. Knoten werden dann als gleich angesehen, wenn sie über die Abbildung M aufeinander abgebildet werden. Aus den gleichen Gründen wie der Ansatz aus Abschnitt 2.2.3.1 ist die δ-Ähnlichkeit nicht weiter relevant.

2.2.3.3 Ähnlichkeit von Prozessmatrizen

Die von Bae et al. (2006b) vorgeschlagene Ähnlichkeit via Prozessmatrizen ist im Grunde ein Abgleich der in einem Modell vorhandenen Kanten. Zunächst wird die Obermenge der in beiden zu vergleichenden Modellen $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ vorhandenen Knoten bestimmt. Gleiche Knoten sind hierbei wieder über die zugrunde liegende Abbildung definiert. Dann wird für beide Prozessmodelle separat eine quadratische Matrix PM_k gebildet, $k = 1, 2$, für die gilt, dass $PM_k(i, j) = 1$, falls $(n_{k,i}, n_{k,j}) \in E_k$, und $PM_k(i, j) = 0$ sonst. Es wird also für jedes Modell eine Matrix angelegt und eingetragen, zwischen welchen Knoten eine Kante existiert. Anschließend wird die elementweise Differenz beider Matrizen berechnet, um Diskrepanzen zwischen den beiden Modellen zu entdecken. Die Einträge dieser Differenzmatrix DM können -1, 0 oder 1 sein, also $(PM_1 - PM_2)(i, j) \in \{-1, 0, 1\}$. Um aus der Differenzmatrix $DM = PM_1 - PM_2$ ein Distanzmaß zu generieren, wird das Skalarprodukt, genauer: das Frobenius-Skalarprodukt (Horn und Johnson, 2012), von DM mit ihrer Transponierten DM^t berechnet: $\langle A, A^t \rangle = tr(AA^t) \geq 0$, wobei tr die Spur, also die Summe der Diagonalelemente einer Matrix, bezeichnet. Dieses Distanzmaß erfüllt die Eigenschaften einer Metrik. Es ist kein Ähnlickeitsmaß, kann allerdings durch geeignete Normierung auf $[0, 1]$, zum Beispiel mit Gleichung (2.2), in ein solches umgewandelt werden. Dies wird von Bae et al. (2006b) allerdings nicht ausgeführt.

2.2.3.4 Anteil gleicher Knoten und Kanten

Um die Ähnlichkeit von Prozessmodellvarianten bezogen auf einen Vergleichsprozess zu bestimmen, schlagen Minor et al. (2007) einen relativ unkomplizierten Ansatz vor: Da es sich bei den betrachteten Modellen um Varianten eines Vergleichsprozesses handelt, dürfen korrespondierende Aktivitäten immer als identische Aktivitäten aufgefasst werden. Die Abbildung M ist somit die Identitätsabbildung und ein Knoten hat entweder eine identische Entsprechung im anderen Modell oder gar keine Entsprechung. Identische Knoten werden miteinander identifiziert, sodass Mengenoperationen auf den Mengen der Knoten und Kanten jeweils problemlos durchgeführt werden können. Abgebildete bzw. identische Kanten werden wie in Definition 2.7 festgelegt. Die Ähnlichkeit zweier Prozessmodelle ergibt sich dann mit dem Ansatz von Minor et al. (2007) über den Anteil gleicher Knoten und gleicher Kanten. Diese Berechnung ist explizit nur für Prozessmodelle ohne Kontrollflussverzweigungen konstruiert, da nur Aktivitäten, für die bestimmt werden kann, ob es eine Entsprechung im anderen Modell gibt oder nicht,

und Kanten zwischen Aktivitäten berücksichtigt werden:

$$fsim_{parts}(G_1, G_2) = 1 - \frac{|N_1 \setminus N_2| + |N_2 \setminus N_1| + |E_1 \setminus E_2| + |E_2 \setminus E_1|}{|N_1| + |N_2| + |E_1| + |E_2|} \in [0, 1]$$

Um die von Kontrollflussverzweigungen gegebenen Informationen im Prozessmodell beim Vergleich ebenfalls zu nutzen, d. h. insbesondere Prozessmodelle nach Definition 1.1 mit der Verfeinerung der Knoten in Gleichung (1.1) zuzulassen, schlagen Minor et al. (2007) zwei verschiedene Wege vor. Anstatt Verzweigungen einfach auszublenden (Methode 1), also zwei Kanten (n_1, c) und (c, n_2) durch eine Kante (n_1, n_2) zu ersetzen, wobei $n_1, n_2 \in N$ und $c \in C$, siehe Abbildung 2.7, werden die durch das Ausblenden neu entstehenden Kanten zwischen Aktivitäten entsprechend der ausgeblendeten Elemente beschriftet (Methode 2). Kanten erhalten also ebenso eine Beschriftung, sodass die Funktion λ um die Kanten als Eingabemenge erweitert werden muss. Zwei Kanten sind dann gleich, wenn ihr Anfangs- und Endknoten sowie die Beschriftungen jeweils gleich sind. Ein Beispiel ist in Abbildung 2.8 dargestellt. Eine weitere Art, den Kontrollfluss zu berücksichtigen, ist die, für jeden Typ an Kontrollflussknoten, der im Modell vorkommt, einen Stellvertreterknoten zu benennen, unabhängig davon, wie oft der Kontrollflussknotentyp im Modell tatsächlich auftritt (Methode 3). Die Kanten werden dann entsprechend angepasst, sodass nur noch Aktivitäten und Stellvertreterknoten miteinander verbunden sind, siehe Abbildung 2.9. Egal welche Methode verwendet wird, kann immer der Anteil gleicher Knoten und Kanten als Ähnlichkeitsmaß verwendet werden.

Insbesondere Methode 2, die als Verfeinerung von Methode 1 angesehen werden kann, wird in abgewandelter Form in Abschnitt 4.3 aufgegriffen. Anstatt dass Kanten eine zusätzliche Beschriftung erhalten, wird dort die durch die ausgeblendeten Gateways implizierte Information den Aktivitäten des Prozessmodells zugeschrieben. Methode 3 ist aufgrund der Umstrukturierung, die andere Aspekte der Modelle wie beispielsweise den Anteil der abgebildeten Kanten beeinflussen kann, für diese Arbeit nicht geeignet.

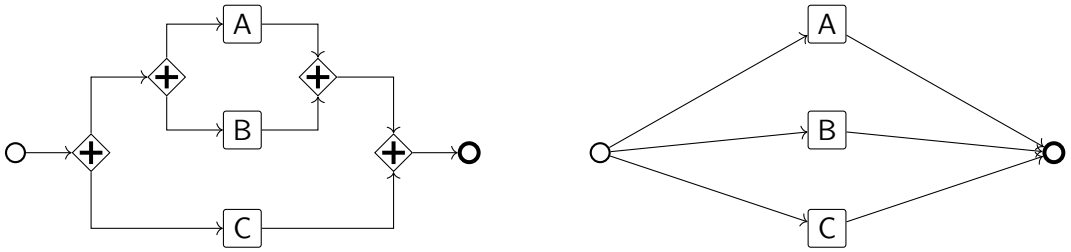


Abbildung 2.7: Abstraktion eines Prozessmodells nach Methode 1.

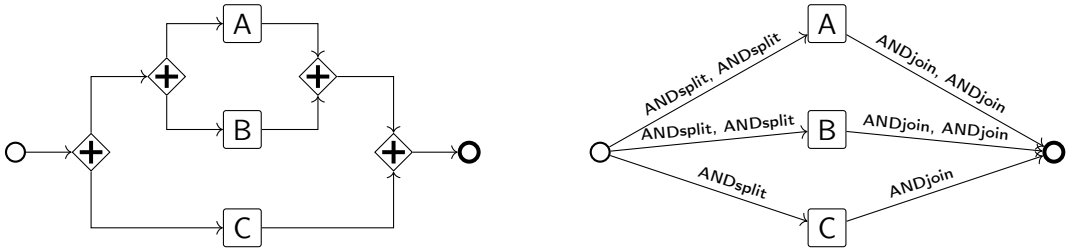


Abbildung 2.8: Abstraktion eines Prozessmodells nach Methode 2.

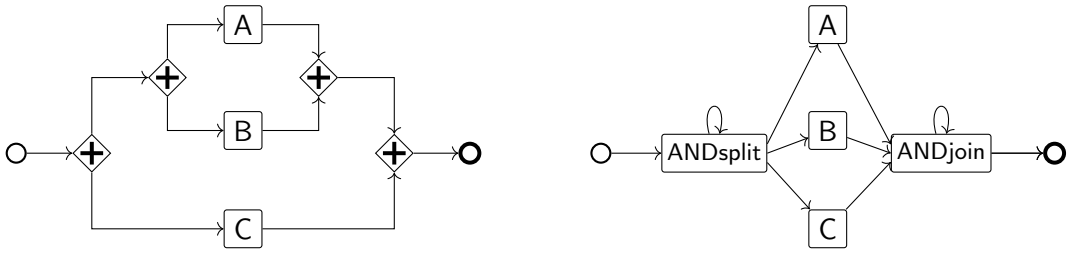


Abbildung 2.9: Abstraktion eines Prozessmodells nach Methode 3.

2.2.3.5 Kontextbasierte Ähnlichkeit mit separater Abbildung

Von Dijkman et al. (2011) wird die kontextbasierte Ähnlichkeit vorgeschlagen, mit der die lokale Struktur von Prozessmodellen berücksichtigt wird. Dijkman et al. (2011) definieren ihr Ähnlichkeitsmaß zwar auf EPKs, die Unterscheidung zwischen Funktionen und Ereignissen, wie sie in EPKs gilt, ist aber nicht zwingend notwendig für die Definition des Maßes, weswegen die kontextbasierte Ähnlichkeit auch auf den abstrakten Prozessgraphen aus Definition 1.1 berechnet werden kann.

Es wird zunächst der Begriff des Pfades eingeführt: Ein Pfad ist ein Tupel an direkt verbundenen Kanten. Zwischen den Knoten n_1 und n_k des Prozessmodells $G = (N, E, \lambda)$ existiert ein Pfad, falls $(n_1, n_2), (n_2, n_3), \ldots, (n_{k-1}, n_k) \in E, n_i \in N \ \forall i = 1, \ldots, k$. Hier ist insbesondere auch der leere Pfad, das ist der Pfad von einem Knoten auf sich selbst, enthalten. Ein besonderer Pfad ist der Verzweigungspfad: Zwischen den Aktivitäten n_1 und n_k, $n_1 \neq n_k$, existiert ein Verzweigungspfad, falls n_1 und n_k nur über Verzweigungsknoten miteinander verbunden sind, ohne dass andere Aktivitäten dazwischen liegen, d. h. $(n_1, n_2), (n_2, n_3), \ldots, (n_{k-1}, n_k) \in E, n_1, n_k \in A$ und $n_2, \ldots, n_{k-1} \in C$ mit $N = A \cup C$. Es wird dann $n_1 \overset{c}{\hookrightarrow} n_k$ geschrieben. Verzweigungspfade, die über eine Schleife eine Aktivität mit sich selbst verbinden, sind hier nicht erlaubt. Es sei angemerkt, dass zwischen n_1 und n_k auch gar kein anderer Knoten liegen kann, dass n_1 und n_k also direkt miteinander verbunden sind ($(n_1, n_k) \in E$). Für jeden Knoten kann somit der Eingangs- und Ausgangskontext bestimmt werden: $n^{in} = \{n' \in A \mid n' \overset{c}{\hookrightarrow} n\}$ ist der Eingangskontext von Aktivität n, das sind alle vorhergehenden Aktivitäten, die direkt oder höchstens über dazwischenliegende Gateways mit n verbunden sind. Entsprechend ist $n^{out} = \{n' \in A \mid n \overset{c}{\hookrightarrow} n'\}$ der Ausgangskontext, also alle direkt oder über Gateways verbundenen Nachfolgeaktivitäten von n.

Wird nun die kontextbasierte Ähnlichkeit von Knoten n_1 aus G_1 und Knoten n_2 aus G_2 bestimmt, so wird die Ähnlichkeit der Eingangs- und Ausgangskontexte von n_1 und n_2 betrachtet. Es sei $M^{in*}_{sim} : n_1^{in} \to n_2^{in}$ die beste partiell injektive Abbildung zwischen dem Eingangskontext von n_1 und dem von n_2, wobei für die Bestimmung der besten Abbildung ein Ähnlichkeitsmaß sim auf den Knoten der Eingangskontexte benötigt wird, beispielsweise sim_{sed} oder sim_{ling}. Die Abbildung M^{in*}_{sim} ist dann die beste Abbildung, wenn gilt $\sum_{(n',n'') \in M^{in*}_{sim}} sim(n', n'') \geq \sum_{(n',n'') \in M^{in}_{sim}} sim(n', n'')$ für beliebige Abbildungen M^{in}_{sim} auf den Eingangskontexten von n' und n'' mit festgelegtem Ähnlichkeitsmaß sim. Analog dazu ist M^{out*}_{sim} die beste partiell injektive Abbildung auf den jeweiligen Ausgangskontexten.

Die kontextbasierte Ähnlichkeit ist dann wie folgt definiert:

$$sim_{con}(n_1, n_2) = \frac{|M^{in*}_{sim}|}{2\sqrt{|n_1^{in}|}\sqrt{|n_2^{in}|}} + \frac{|M^{out*}_{sim}|}{2\sqrt{|n_1^{out}|}\sqrt{|n_2^{out}|}} \in [0, 1]$$

Diese kontextbasierte Ähnlichkeit hängt nicht von den konkreten Ähnlichkeitswerten der Elemente der Eingangs- und Ausgangskontexte ab, sondern benötigt lediglich die Anzahl der Abbildungselemente der besten Abbildungen auf den beiden Mengenpaaren (siehe Zähler bei der Berechnung von sim_{con}). Die Übertragung dieses Ansatzes auf M:N-Abbildungen und insbesondere die Anpassung der Abbildung der Eingangs- und Ausgangskontexte ist wegen eventuellen Überschneidungen dieser schwer möglich, weshalb dieser Ansatz im weiteren Verlauf nicht berücksichtigt wird.

2.2.3.6 Kontextbasierte Ähnlichkeit ohne separate Abbildung

Für die kontextbasierte Ähnlichkeit, die von La Rosa et al. (2010) genannt wird, wird keine neue Abbildung M^{in*} oder M^{out*} benötigt, sondern mit der Abbildung $M : G_1 \rightarrow G_2$ gearbeitet. Die Definitionen der Eingangs- und Ausgangskontexte werden wieder verwendet, allerdings erlauben La Rosa et al. (2010) die Definition der Eingangs- und Ausgangskontexte auch für Verzweigungsknoten. Das bedeutet, wenn Verzweigungsknoten aufeinander abgebildet werden, kann die Anzahl der vorhergehenden bzw. nachfolgenden Aktivitäten der Gateways verglichen werden.

$$sim_{con'}(n_1, n_2) = \frac{|M(n_1^{in}) \cap n_2^{in}| + |M(n_1^{out}) \cap n_2^{out}|}{\max(|n_1^{in}|, |n_2^{in}|) + \max(|n_1^{out}|, |n_2^{out}|)} \in [0, 1]$$

Eine höhere Ähnlichkeit ergibt sich für das Maß $sim_{con'}$ tendenziell dann, wenn von den Gateways Splits aufeinander abgebildet werden und Joins aufeinander abgebildet werden. Jedoch erlaubt $sim_{con'}$ nicht, die Typen der Gateways, also Parallelität (AND) und Exklusivität (XOR), zu unterscheiden. Falls in der 1:1-Abbildung gemäß Definition 2.2 ausschließlich Aktivitäten aufeinander abgebildet werden, wird die Aussagekraft dieses Maßes stark eingeschränkt, da jede Aktivität einen Eingangs- und Ausgangskontext von Mächtigkeit 1 hat. Obwohl dieser Ansatz grundsätzlich einfacher in Konzeption und einer möglichen Umsetzung als der aus Abschnitt 2.2.3.5 ist, spielt er wegen des genannten Nachteils für den weiteren Verlauf der Arbeit keine Rolle.

2.2.3.7 Merkmalsbasierte Ähnlichkeitsbestimmung

Die Verwendung von Merkmalen (*features*) ist eine gängige Methode des maschinellen Lernens, um Objekte auf eine abstrahierte Art und Weise beschreiben zu können. Auch Prozessmodelle können auf bestimmte Merkmale reduziert werden, anhand derer dann eine Ähnlichkeit berechnet wird (Yan et al., 2010). Ein erstes von Yan et al. (2010) genanntes Merkmal ist die Beschreibung der Aktivitäten. Auf dieses Merkmal können Ähnlichkeitsbestimmungen, wie beispielsweise in Abschnitt 2.2.2 beschrieben, angewendet werden. Ein zweites, repräsentatives Merkmal eines Prozessmodells ist, zusätzlich zur Beschreibung der Aktivitäten, dessen Struktur, auf Basis derer in Abschnitt 2.2.3 einige Abgleichsmöglichkeiten vorgestellt werden. Um die Struktur für die weiteren Berechnungen effizient verwenden zu können, wird diese von Yan et al. (2010) zu den folgenden Strukturmerkmalen abstrahiert:

- Startmerkmal: Ein Knoten ohne Vorgänger
- Stoppmerkmal: Ein Knoten ohne Nachfolger
- Sequenzmerkmal der Länge s, $s \geq 2$: Eine Kette an $s - 1$ zusammenhängenden Kanten

- Verzweigungsmerkmal der Größe s, $s \geq 3$: Ein Knoten und seine $s-1$ direkten Nachfolger
- Vereinigungsmerkmal der Größe s, $s \geq 3$: Ein Knoten und seine $s-1$ direkten Vorgänger

Weiter wird das sogenannte Rollenmerkmal eines Knotens definiert. Jeder Knoten nimmt mindestens eine der folgenden Rollen ein: Start, Stopp, Verzweigung, Vereinigung und regulärer Knoten. Auch diese Rollen werden über die Anzahl der Vorgänger-/Nachfolgerknoten bestimmt. So ist zum Beispiel ein Vereinigungsknoten ein Knoten mit mindestens zwei Vorgängern. Hat er auch mehr als einen Nachfolgerknoten, so ist er gleichzeitig ein Verzweigungsknoten. Eine Rollenähnlichkeit zwischen zwei Knoten erfolgt dann über die Betrachtung der Anzahl an unterschiedlichen Vorgänger-/Nachfolgerknoten. Für Knoten mit bestimmten Rollenkombinationen ist jeweils eine eigene Berechnungsvorschrift gegeben. Zum Beispiel wird für zwei Knoten n und m, die beide weder die Rolle Start noch die Rolle Stopp innehaben, die Ähnlichkeit folgendermaßen berechnet:

$$rsim(n,m) = \frac{1}{2} \cdot \left(\left(1 - \frac{|(|n \bullet| - |m \bullet|)|}{|n \bullet| + |m \bullet|} \right) + \left(1 - \frac{|(|\bullet n| - |\bullet m|)|}{|\bullet n| + |\bullet m|} \right) \right)$$

Hierbei bezeichnet $|n \bullet|$ die Anzahl der Nachfolgerknoten von n und $|\bullet m|$ die Anzahl der Vorgängerknoten von m. Um Rollen, die zu häufig in einem Modell auftauchen, nicht zu berücksichtigen, da ihre Ähnlichkeitswerte laut Yan et al. (2010) wenig aussagekräftig sind, wird von den Autoren ein Schwellenwert eingefügt, der nur eine maximale relative Häufigkeit für die Rollen zulässt. Häufigere Rollen werden ignoriert.

Mit Hilfe aller genannten Merkmale wird dann ein Merkmalsabgleich durchgeführt. Der Knotenmerkmalsabgleich betrachtet dabei die Label und das Rollenmerkmal und führt eine Fallunterscheidung durch. Ist die Ähnlichkeit der Label zweier Knoten hoch genug, das heißt über einem relativ hoch angesetzten Schwellenwert, so werden die Knoten sofort als ähnlich eingestuft. Sind die Labels sich so ähnlich, dass sie einen tiefer angesetzten Schwellenwert überschreiten, nicht jedoch den hohen Schwellenwert, so wird ihre Rollenähnlichkeit berechnet. Ist diese über einem gegebenen Level, werden auch diese Knoten als ähnlich eingestuft.

Der Strukturmerkmalsabgleich verwendet die Ähnlichkeit der Knoten. Während die Knoten mit Start- und Stoppmerkmalen jeweils direkt miteinander abgeglichen werden, ist eine Sequenzähnlichkeit nur dann gegeben, wenn die Sequenzen gleiche Länge haben und die Knoten der beiden Sequenzen jeweils ähnlich zueinander sind unter Einhaltung der Sequenzreihenfolge. Verzweigungs- und Vereinigungsähnlichkeit sind wie folgt definiert: Zwei gleich große Verzweigungen sind sich ähnlich, wenn die jeweiligen Verzweigungsknoten ähnlich sind und je zwei der direkten Nachfolgerknoten sich ähneln; analog gilt Ähnlichkeit von Vereinigungen mit Vorgängern statt Nachfolgern. Der Strukturmerkmalsabgleich basiert dann, im Grunde genommen, auch auf einem Abgleich der Label. Ohne einen Labelabgleich kann kein Strukturabgleich durchgeführt werden.

Es seien G_1 und G_2 zwei Prozessmodelle. Es bezeichne F_1 die Menge aller Merkmale von G_1 und F_2 die Menge aller Merkmale von G_2. Die Funktion $match(\cdot,\cdot)$ gibt an, ob sich die beiden Prozessmodelle im jeweils betrachteten Merkmal ähnlich genug sind. Die merkmalsbasierte Ähnlichkeit sim_f zwischen G_1 und G_2 errechnet sich aus

$$\begin{aligned} fsim_f(G_1,G_2) &= \frac{|\{f_1 \in F_1 \mid \exists f_2 \in F_2 : match(f_1,f_2)\}| + |\{f_2 \in F_2 \mid \exists f_1 \in F_1 : match(f_1,f_2)\}|}{|F_1| + |F_2|} \\ &\in [0,1]. \end{aligned}$$

Eine Abbildung zwischen den beiden verglichenen Prozessmodellen im Ganzen ist mit diesem Verfahren nicht gegeben, allerdings wird dies von Yan et al. (2010) auch nicht als Ziel der Arbeit angegeben. Dieses sei in erster Linie eine wenig rechenintensive Schätzung der Ähnlichkeit, die für zweifelhafte Modelle eine detailliertere Ähnlichkeitsberechnung nach sich ziehen kann.

Das grundsätzliche Vorgehen, den Aktivitäten Eigenschaften, die sich aus der Modellstruktur ergeben, zuzuweisen, wird in Abschnitt 4.3 aufgegriffen, da ein solches Vorgehen, wie Yan et al. (2010) schreiben, eine deutliche Vereinfachung bedeutet, was die Rechenzeit abgelangt. Es werden in Abschnitt 4.3 jedoch andere Merkmale verwendet. Außerdem wird das Weglassen häufiger Merkmale, was ein Weglassen von Information bedeutet, nicht übernommen. Die Höhe eines Schwellenwerts festzulegen stellt einen großen Unsicherheitsfaktor bei der Ähnlichkeitsbestimmung dar.

2.2.3.8 Ähnlichkeit von Prozessmodellblöcken

Von Bae et al. (2006a) werden Prozessmodelle von ihren Abhängigkeitsgraphen repräsentiert betrachtet. Ein Abhängigkeitsgraph (vgl. auch Bae et al., 2006b) ist ein Tupel (DN, DE), wobei in DN die Aktivitäten und in DE die Kanten eines Modells gespeichert werden. Jede Aktivität ist wiederum ein Tupel (NT, NN, TC, NS), wobei NT den Knotentyp, NN den Knotenname, TC die Auslösebedingung und NS den Status des Knotens, geschaltet oder nicht geschaltet, beschreibt. Eine Kante ist ebenfalls ein vierstelliges Tupel $(EN, DP_{nd}, AV_{nd}, ES)$, wobei EN der Kantenname, DP_{nd} der Startknoten, AV_{nd} der Zielknoten und ES der Status der Kante, ausgelöst oder nicht ausgelöst, ist. Da der Abhängigkeitsgraph vornehmlich den Datenfluss und die daraus resultierenden Abhängigkeiten repräsentiert, sind Kanten von einem Knoten auf denselben Knoten nicht erlaubt, genauso wenig wie isolierte Knoten oder Gruppen von Knoten. Da die Informationen über (nicht) geschaltete Knoten bzw. (nicht) ausgelöste Kanten für den Ähnlichkeitsabgleich nicht benötigt werden, kann der Abhängigkeitsgraph analog zur Prozessmodelldefinition aus Definition 1.1 gesehen werden. In Abbildung 2.10, die einen Abhängigkeitsgraph zeigt, sind deswegen diese Informationen auch nicht abgebildet.

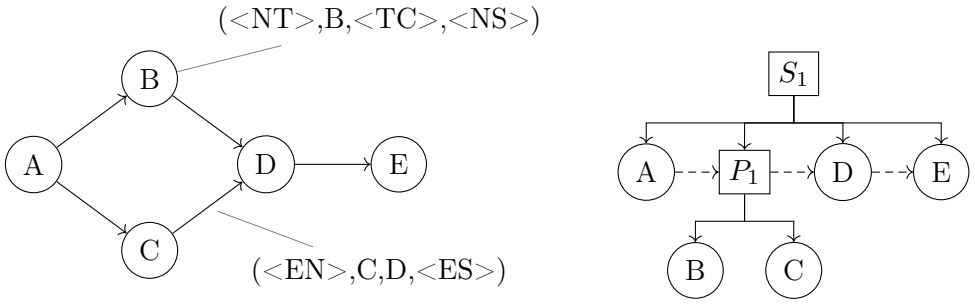


Abbildung 2.10: Beispiel für einen Abhängigkeitsgraphen (links) und einen daraus abgeleiteten Block-Baum (rechts).

Zusätzlich zu Knoten und Kanten ist jeder Abhängigkeitsgraph aus fünf verschiedenen Blockarten aufgebaut: serieller Block, iterativer Block, AND-Block, XOR-Block und OR-Block. Anhand dieser Blöcke kann jeder geschachtelte Abhängigkeitsgraph hierarchisch in Baumstruktur, dem sogenannten Block-Baum, repräsentiert werden, wobei die Wurzel dieses Baumes stets ein serieller Block ist (z. B. S_1 in Abbildung 2.10). Für die Ähnlichkeitswertberechnung wird die Unterscheidung zwischen AND-, XOR- und OR-Blöcken nicht beachtet – es

gibt nur eine Art paralleler Blöcke (z. B. P_1 in Abbildung 2.10), wobei der Begriff „paralleler Block“ nicht nur eine AND-Verzweigung bezeichnet, sondern alle Arten an Verzweigungen mit Ausnahme von Schleifen. Iterative Blöcke (in Beispielabbildung 2.10 keiner vorhanden) werden ebefalls nicht berücksichtigt, da sie die Struktur des Graphen nach Angabe der Autoren nicht beeinflussen, d. h., die Kante im Modell, die zurück zu einem bereits abgehandelten Knoten führt, wird als nicht existent angesehen.

Die Block-Bäume werden anschließend zu normalisierten, binären Bäumen transformiert und zu jedem Baum sein zugehöriger Verzweigungsvektor, der zu allen 3-Tupeln an möglichen Verzweigungen (Wurzel, Kind 1, Kind 2) angibt, ob die jeweilige Verzweigung im Baum auftaucht (Wert 1) oder nicht auftaucht (Wert 0). Abbildung 2.11 zeigt den normalisierten, binären Baum aus Abbildung 2.10 und den zugehörigen Verzweigungsvektor.

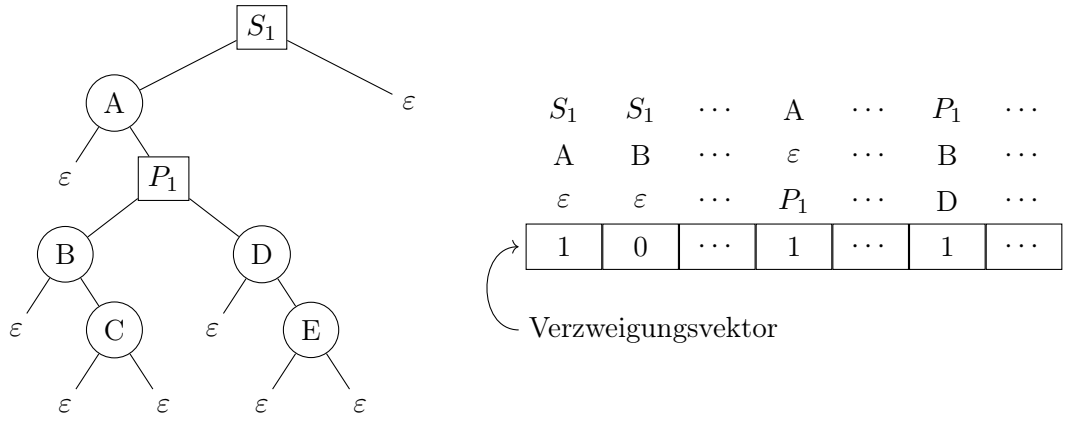


Abbildung 2.11: Beispiel für einen normalisierten, binären Baum (links) und einen daraus gebildeten Verzweigungsvektor (rechts).

Seien G_1 und G_2 die beiden zu vergleichenden Prozessmodelle (Abhängigkeitsgraphen). Mit T_1 bzw. T_2 werden ihre Binärbaumrepräsentationen bezeichnet und mit $BBV(T_1) = (b_{1,1}, b_{1,2}, \ldots, b_{1,|\Gamma|})$ bzw. $BBV(T_2) = (b_{2,1}, b_{2,2}, \ldots, b_{2,|\Gamma|})$ ihre Verzweigungsvektoren, wobei $b_{i,j} \in \{0, 1\}$ und Γ die Menge aller 3-Tupel an möglichen Verzweigungen darstellt. Ein Beispiel ist in Abbildung 2.11 ausschnittsweise gegeben. Die strukturelle Distanz von T_1 und T_2 bzw. von G_1 und G_2 ist dann $d_{bbv}(G_1, G_2) = d_{bbv}(T_1, T_2) = \sum_{j=1}^{|\Gamma|} |b_{1,j} - b_{2,j}| \in [0, |\Gamma|]$. Diese Distanz lässt sich normieren und auf $[0, 1]$ invertieren, was von Bae et al. (2006a) nicht explizit ausgeführt wird, um ein Ähnlichkeitsmaß $fsim_{bbv}$ zu erhalten:

$$fsim_{bbv}(G_1, G_2) = 1 - \frac{1}{|\Gamma|} \sum_{j=1}^{|\Gamma|} |b_{1,j} - b_{2,j}|$$

Die Ähnlichkeitsbestimmung mittels Prozessblöcken stellt eine gute Möglichkeit dar, die Strukturinformationen in den Prozessmodellen bei einem Abgleich zu berücksichtigen. Der Grund, warum dieses Verfahren nicht weiter berücksichtigt werden kann, ist der, dass die Block-Baum-Darstellung nur für einzelne Aktivitäten funktioniert, nicht aber für Aktivitätenmengen und somit nicht auf M:N-Abbildungen übertragbar ist.

2.2.3.9 Kophänetische Distanz

Sánchez-Charles et al. (2016) schlagen die kophänetische Distanz, die ursprünglich für Den-

drogramme, das sind Bäume zur Darstellung von hierarchischen Clustern (Phipps, 1971), entwickelt wurde, zum Vergleich von Prozessmodellen vor. Nach Aussage der Autoren ist es so möglich, sowohl das Verhalten als auch die Struktur von Prozessmodellen beim Vergleich zu berücksichtigen. Dies ist, wenn die Variante mit angepassten Gewichten gewählt wird, zu einem gewissen Anteil wahr. Allerdings beruht die Berücksichtigung des Verhaltens in erster Linie auf subjektiv gewählten Gewichten und deren Differenzen. In größeren Prozessmodellen können die gleichen Differenzen für unterschiedliche Gatewayschachtelungen auftreten, somit sind die Ergebnisse dieser Abgleichsmethode alles andere als eindeutig.

Da die kophänetische Distanz auf gewichteten, gewurzelten Bäumen definiert ist, das sind Bäume mit einer eindeutigen Wurzel, gerichteten Kanten mit Gewichten und benannten Knoten, müssen die Prozessmodelle in einem ersten Schritt in Prozessbäume übersetzt werden. Jede Aktivität des Prozessmodells wird hierbei zu einem Blatt des entsprechenden Prozessbaums, wobei die inneren Knoten die Schachtelung der Aktivität im Prozessmodell, also das Verhalten, widerspiegeln. Erlaubte Verhaltensmuster sind Sequenzen (SEQ), parallele Verzweigungen (AND), exklusive Verzweigungen (XOR) und Schleifen (LOOP), wobei die Schleifen keine Aktivitäten auf dem rückwärts zeigenden Zweig beinhalten dürfen. Abbildung 2.12 zeigt eine beispielhafte Transformation eines Prozessmodells in einen Prozessbaum, bei dem die Tiefe der Knoten jeweils im Index angegeben ist. Der kophänetische Wert zweier Blätter ist die Tiefe des am tiefsten gelegenen, gemeinsamen Vorfahren. Im äußersten Fall ist dies der Wurzelknoten selbst. Für jedes Aktivitätenpaar eines Modells kann so ein kophänetischer Wert bestimmt werden, der in den kophänetischen Vektor, eigentlich eine Dreiecksmatrix, eingetragen wird. Ein Beispiel für solch eine Matrix ist ebenfalls in Abbildung 2.12 gegeben.

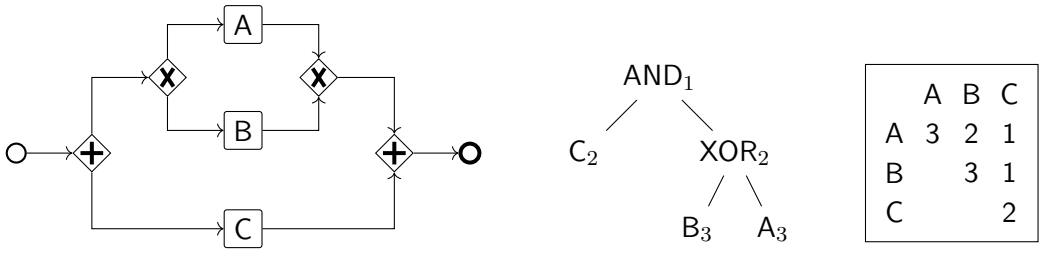


Abbildung 2.12: Prozessmodell (links), Prozessbaum (Mitte) und kophänetischer Vektor (rechts).

Die Berechnung der kophänetischen Distanz zweier Prozessmodelle setzt eine 1:1-Abbildung zwischen den beiden Modellen voraus. Zum Abgleich der Modelle können deren kophänetische Vektoren, für deren Berechnung implizit eine Kantengewichtung von 1 für jede Kante vorausgesetzt wird, herangezogen werden. Eine Distanzmatrix enthält die betragsmäßigen Differenzen der einzelnen Einträge, die dann zu einem Distanzwert aufsummiert werden. Einen detaillierteren Vergleich erhält man, wenn man die Gewichte der Pfade zu den Prozessbaumelementen einzeln für jedes der vier erlaubten Verhaltensmuster vergibt. So werden von Sánchez-Charles et al. (2016) für Kanten von einem XOR-Knoten aus ein Gewicht von 0,5 statt 1 vorgeschlagen und für eine Kante von einem SEQ-Knoten aus ein Gewicht, das der Tiefe des tiefsten bereits besuchten Kindes eines Geschwisterknotens des SEQ-Knotens plus 1 entspricht. Kinder eines LOOP-Startknotens erhalten als Gewichtsbonus die maximale Tiefe des der Schleife zugrunde liegenden Teilbaums. Das Beispiel aus Abbildung 2.12 ändert sich damit wie in Abbildung 2.13 gezeigt ab, wobei bei AND-Knoten keine Änderung der Kantengewichte vorgenommen wird. Abbildung 2.14 zeigt ein Beispiel für die differenzierte

Gewichtung von Sequenz- und Schleifenknoten.

Der Distanzwert, der auf diese Weise gewonnen wird, muss noch in ein Ähnlichkeitsmaß umgerechnet werden, um im vierstufigen Ansatz aus Abschnitt 2.2.1 verwendet werden zu können. Allerdings ist selbst bei einer Umrechnung auf einen normierten Wert zwischen 0 und 1 die Interpretation des Ähnlichkeitswertes schwierig, besonders dann, wenn die separate Gewichtung der Kanten vorgenommen wird. Sie erlaubt zwar, Unterschiede zwischen beiden Modellen relativ genau aufzuzeigen, doch auf eine vor allem für Menschen und nicht für Maschinen lesbare Art und Weise (Sánchez-Charles et al., 2016). Dies macht eine Automatisierung des Ansatzes schwierig.

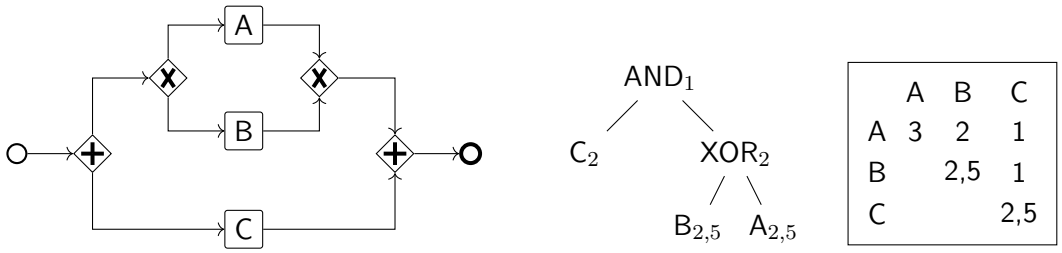


Abbildung 2.13: Prozessmodell (links), Prozessbaum mit angepassten Gewichten (Mitte) und kophänetischer Vektor (rechts).

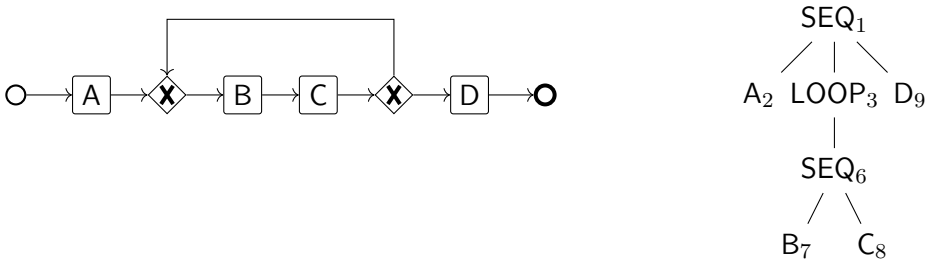


Abbildung 2.14: Prozessmodell (links) und Prozessbaum (rechts) mit angepassten Gewichten für Sequenz und Schleife.

2.2.3.10 Ansätze aus dem Graph Matching

Aus dem weiten Feld des Graph Matchings lassen sich viele Ansätze auf einen Ähnlichkeitsabgleich von Prozessmodellen übertragen, wobei sich diese Ansätze typischerweise auf die Struktur der Prozessmodelle beziehen und die Ausführungssemantik außer Acht lassen. Die wichtigsten Konzepte des Graph Matching sind

- Graphisomorphismen,
- Teilgraphisomorphismen,
- die Bestimmung des größten, gemeinsamen Teilgraphen und
- der Editierabstand von Graphen, der gegenüber Abweichungen, im Gegesatz zum Test auf Isomorphie, toleranter ist (Bunke, 2000).

Viele der in diesem Gebiet vorgestellten Abgleichsmethoden stellen eine Ähnlichkeitsmetrik zur Verfügung, also ein Ähnlichkeitsmaß, das unter anderem die Dreiecksungleichung (siehe Abschnitt 2.1) erfüllt, was gerade für den paarweisen Abgleich von vielen verschiedenen Modellen in großen Repositorien hilfreich sein kann. Die Ansätze des Graph Matchings sind diejenigen, die auch die Grundlage des Abgleichs von Prozessmodellen mit M:N-Abbildungen darstellen.

Graphisomorphismus, Teilgraphisomorphismus und größter gemeinsamer Teilgraph Ein Isomorphismus, also eine nachbarschaftserhaltende, bijektive Abbildung, zwischen zwei gerichteten Graphen ist wie folgt definiert:

Definition 2.10 (Isomorphismus)**.** Seien $G_1 = (N_1, E_1)$ und $G_2 = (N_2, E_2)$ zwei Graphen (Prozessmodellgraphen). Eine bijektive Abbildung $f : N_1 \to N_2$ heißt Isomorphismus zwischen G_1 und G_2, wenn

$$(n, m) \in E_1 \Leftrightarrow (n', m') \in E_2 \; \forall (n, m) \in E_1, \forall (n', m') \in E_2 : n' = f(n), m' = f(m).$$

Sind die Graphen zweier Prozessmodelle isomorph zueinander, d. h., lässt sich ein Isomorphismus zwischen den beiden Graphen finden, so kann man sagen, dass sich die beiden Modelle – zumindest in ihrer Struktur – gleichen. Isomorphismen haben jedoch den Nachteil, dass sie bijektiv sind, die Prozessmodelle also die gleiche Anzahl an Knoten haben müssen, damit man überhaupt einen Isomorphismus finden kann (Fortin, 1996). Ist eine Abbildung M zwischen den zu vergleichenden Prozessgraphen hergestellt, muss also die Isomorphismuseigenschaft überprüft werden, d. h., es wird getestet, ob die gewählte Abbildung nachbarschaftserhaltend ist. Es muss gelten:

$$\forall (n, m) \in E_1 \Rightarrow (M(n), M(m)) \in E_2 \; \wedge \; \forall (n', m') \in E_2 \Rightarrow (M^{-1}(n'), M^{-1}(m')) \in E_1$$

Kann kein Isomorphismus zwischen zwei Graphen gefunden werden, lässt sich jedoch nichts über den Grad der Unähnlichkeit der Prozessmodelle aussagen.[5]

Haben die zu vergleichenden Prozessmodellgraphen unterschiedliche Knotenanzahl, so kann man auf Teilgraphenisomorphie (Monomorphie) (siehe Shapiro und Haralick, 1981) testen. Hier wird überprüft, ob ein Isomorphismus zwischen dem kleineren Graphen und einem Subgraphen des größeren Graphen besteht (Ullmann, 1976). Kann kein Teilgraphenisomorphismus gefunden werden, so besteht auch die Möglichkeit, nach dem größten gemeinsamen, d. h., isomorphen, Teilgraphen zu suchen (Bunke und Jiang, 2000).

Editierabstand von Graphen Ähnlich wie der Editierabstand für Zeichenketten (siehe Abschnitt 2.2.2.1) kann auch für Graphen ein Editierabstand definiert werden (Bunke, 1997). Jede Änderung e am Graphen – das Einfügen eines neuen Knotens bzw. einer neuen Kante, das Löschen eines Knotens bzw. einer Kante, das Ändern eines Knotenlabels bzw. eines Kantenlabels, was dem Ersetzen eines Knotens bzw. einer Kante entspricht – wird mit gewissen Kosten $c(e) \in [0, 1]$ versehen. Es ist zu beachten, dass das Löschen eines Knotens das Löschen der an diesem Knoten anhaftenden Kanten nach sich zieht und dass eine Kante nur zwischen bestehenden Knoten eingefügt werden kann. Haben verschiedene Operationen unterschiedliche Kosten, so spricht man auch von einem gewichteten Editierabstand. Eine Folge

[5] Da eine partiell injektive Abbildung eingeschränkt auf Definitions- und Wertemenge bijektiv ist, sind Homomorphismen äquivalent zu Isomorphismen.

von Änderungsoperationen $(e_1, e_2, \ldots, e_k)$, die einen Graphen G_1 in einen anderen Graphen G_2 transformieren – kurz: $G_2 = e_k(\ldots e_2(e_1(G_1))\ldots)$ – hat somit eine bestimmte Menge an Kosten. Je geringer diese Kosten sind, desto geringer ist der Unterschied zwischen beiden Graphen, d. h., desto ähnlicher sind sich beide Graphen. Für den Editierabstand δ_{ged} wird über alle Sequenzen, die einen Graphen G_1 in einen anderen Graphen G_2 überführen, bezüglich ihrer Kosten minimiert:

$$\delta_{ged}(G_1, G_2) = \min_{\substack{k\in\mathbb{N}_0,\\ e_1,\ldots,e_k}} \left\{ \sum_{i=1}^{k} c(e_i) \mid G_2 = e_k(\ldots e_2(e_1(G_1))\ldots) \right\}$$

Aus diesem Abstandsmaß, das unter bestimmten Voraussetzungen an die Kostenfunktion sogar eine Metrik ist (Bunke und Jiang, 2000), lässt sich eine Ähnlichkeit herleiten, die Editierähnlichhkeit für Graphen (*graph-edit similarity*; ges), beispielsweise über $sim_{ges} = 1/1+\delta_{ged}$.

Werden für die Bestimmung des Editierabstands bzw. der Editierähnlichkeit von Graphen nicht nur gleiche und ungleiche Knoten unterschieden, sondern auch ähnliche Knoten, dann entspricht das Vorgehen dem aus Abschnitt 2.2.1 und es muss auf Basis der Änderungskosten, die mit dem Ähnlichkeitswert negativ korreliert sind, eine beste Abbildung gefunden werden, sodass die Änderungskosten minimiert werden.

Beispielsweise wird der Editierabstand von Graphen zur Ähnlichkeitsbestimmung von Kunze und Weske (2011) in ihrem Framework zur effizienten Suche von ähnlichen Prozessmodellen in großen Repositorien herangezogen. Auch Grigori et al. (2010) verwenden den Editierabstand von Graphen, um Ähnlichkeit von BPEL-Prozessen, die in Prozessgraphen transformiert werden, zu bestimmen. Jedoch werden hier noch zwei zusätzliche Änderungsoperationen zugelassen, nämlich das Aufspalten bzw. das Vereinen zweier Knoten. Das Vereinen ist nur bei benachbarten, sequentiell angeordneten Knoten möglich bzw. das Aufspalten führt immer zu benachbarten, sequentiellen Knoten. Wird eine solche Änderung wiederholt angewendet, ist es mit diesem Ansatz auch möglich, 1:N-Abbildungen zwischen zwei Prozessgraphen zu erzeugen. Bei 1:N-Abbildungen wird ein einzelner Knoten auf eine Menge an Knoten abgebildet. Auf 1:N-Korrespondenzen wird jedoch nur getestet, wenn 1:1-Abbildungen keine akzeptablen Ergebnisse liefern. Des Weiteren können 1:N-Korrespondenzen nur für grundlegende synchrone und asynchrone *invoke*- und *receive*-Aktivitäten (Aufrufen eines Webservice bzw. Anbieten einer Webserviceschnittstelle) der BPEL-Sprache entdeckt werden, nicht für allgemeine Aktivitäten. Die Aufspaltung bzw. Vereinigung erfolgt probeweise und es wird getestet, ob nach der Aufspaltung bzw. Vereinigung ein (besserer) Match vorhanden ist. Der M:N-Ansatz, der in Kapitel 4 der vorliegenden Arbeit vorgestellt wird, benötigt keinen Test auf die Güte einer 1:1-Abbildung sondern lässt von vornherein eine Abbildung von Mengen zu, die auch nicht auf bestimmte Typen von Knoten beschränkt ist.

Es lässt sich feststellen, dass für eine Ähnlichkeitsbestimmung auf Basis der Prozessmodellstruktur eine große Vielfahlt an Methoden besteht. Wie an vielen Stellen erwähnt ist ein Großteil der Methoden jedoch nicht auf M:N-Abbildungen übertragbar. Außerdem sind diejenigen Methoden, die die Graphenstruktur insbesondere unter Verwendung der Kanten benutzen, für einen Abgleich von deklarativen Prozessmodellen nicht anwendbar, auch nicht unter 1:1-Abbildungen. Mehr zu deklarativen Prozessmodellen findet sich in Abschnitt 4.5. Einschränkungen der Anwendbarkeit der vorgestellten Methoden für imperative Prozessmodelle gibt es kaum. Die kophänetische Distanz (Abschnitt 2.2.3.9) lässt nur Loops ohne Aktivitäten auf dem rückwärtigen Kontrollfluss zu. Außerdem stellen diese Methode, wenn sie mit der Kantengewichtung verwendet wird, und die Methode der Prozessmatrizen (Abschnitt 2.2.3.3)

keine Ähnlichkeitsmaße im Sinne von Definition 2.1 zur Verfügung. Über die reine Modellstruktur hinausgehend werden in Abschnitt 2.2.4 Methoden, die das Verhalten eines Prozessmodells berücksichtigen, also beispielsweise die Semantik der verschiedenen Gatewaytypen, vorgestellt.

2.2.4 Verhaltensbasierte Ansätze zum Ähnlichkeitsabgleich

Die verhaltensbasierten Ansätze zum Ähnlichkeitsabgleich betrachten im Unterschied zu strukturbasierten Ähnlichkeitsmaßen nicht den globalen Aufbau von Prozessmodellen, sondern beziehen vielmehr den Sequenzfluss inklusive der Verzweigungsknoten und der damit verbundenen Informationen mit ein. Es ist somit für diese Ansätze in der Regel notwendig, die Knoten eines Prozessmodells anhand Gleichung (1.1) unterscheiden zu können. Manche dieser Ansätze setzen auch voraus, dass sämtliche Ausführungspfade der Prozessmodelle bekannt sind. Dies können unter Umständen unendlich viele sein. Als gleiche Knoten werden wieder diejenigen betrachtet, die die zugrunde liegende Abbildung aufeinander abbildet, d. h., Knoten werden mittels der Abbildung miteinander identifiziert.

Ein erster Ansatz zur Bestimmung der Verhaltensähnlickeit verwendet die Bisimulationseigenschaft (Abschnitt 2.2.4.1), die jedoch eine sehr strenge Auslegung von Ähnlichkeit aufweist. In Abschnitt 2.2.4.2 werden komplette Ausführungspfade zur Ähnlichkeitsbestimmung verwendet, um die Strenge der Bisimulation aufzuheben. Abschnitt 2.2.4.3 arbeitet mit abstrahierten Pfaden bzw. Pfadteilstücken, die grunsätzlich einfacher zu handhaben sind als komplette Ausführungspfade. Ebenfalls eine Abstraktion wird mittels kausaler Fußabdrücke in Abschnitt 2.2.4.4 vorgenommen, wobei hier sogar mehrere Varianten für ein Ähnlichkeitsmaß genannt sind. In den Abschnitten 2.2.4.5 und 2.2.4.6 wird das Verhalten eines Prozessmodells in ein sogenanntes Verhaltensprofil überführt, wobei für diese Überführung ebenfalls Ausführungspfade herangezogen werden.

2.2.4.1 Abgleich mittels Bisimulation

In mehreren Arbeiten, unter anderem von Hidders et al. (2005) und Alves de Medeiros et al. (2008), wird die Ähnlichkeit von Prozessmodellen über die Menge der Ausführungspfade bestimmt. Hierbei können diese Mengen direkt miteinander verglichen werden, oder, wenn sich das Prozessmodell auf ein Transitionssystem, beispielsweise ein Petrinetz, überführen lässt, auch die Eigenschaft der Bisimulation genutzt werden.

Die Bisimulation ist eine Relation, die Zustände von Transitionssystemen in Beziehung setzt, und zwar genau dann, wenn die möglichen Übergänge in den jeweiligen Zuständen dieselben sind. Die Systeme können sich also in den jeweiligen Zuständen gegenseitig simulieren. Hidders et al. (2005) bezeichnen zwei Prozessmodelle als beobachtungsäquivalent, wenn die Mengen ihrer Ausführungspfade die gleichen sind. Dies ist, wie die Autoren zeigen, äquivalent dazu, dass zwei Modelle bisimilar sind. Zusätzlich zur Beobachtungsäquivalenz werden von Hidders et al. (2005) auch noch weitere Äquvalenzdefinitionen genannt, die aber alle, wie auch die Beobachtungsäquivalenz, nur binäre Aussagen über die Ähnlichkeit zweier Modelle treffen. Dies macht die Abgleichsmethoden für den weiteren Verlauf dieser Arbeit wenig interessant.

2.2.4.2 Abgleich der Mengen der Ausführungspfade

Alves de Medeiros et al. (2008) wollen in ihrer Arbeit das Problem der binären Aussage durch einen anderen Ansatz, der ebenfalls auf Ausführungspfaden basiert, lösen. Hierfür wird ein

Eventlog benötigt, der beispielhaftes Verhalten beinhaltet. Diese Beispielausführungen können entweder durch reale Prozessausführung und Logging, durch benutzerdefinierte Szenarien oder durch reine Simulation erzeugt werden. Aus den Pfaden des Eventlogs wird dann die Fitness berechnet, eine Zahl aus $[0, 1]$, die angibt, wie gut der Log zu einem bestimmten Prozessmodell passt. Diese Fitness gibt allerdings nicht unbedingt an, wie ähnlich sich zwei Prozessmodelle sind, denn selbst bei ähnlicher Fitness können die Stellen, an denen die Unterschiede entstehen, verschiedene sein. Um also zwei Modelle miteinander zu vergleichen, wird eine spezielle Definition von Genauigkeit (*precision*) und Trefferquote (*recall*) angegeben. Hierbei wird nicht nur auf Überlappungen in den Pfaden oder gleiche Teilstücke in den Pfaden geachtet, sondern auch die Menge an Aktivitäten, die nicht ausgeführt wurden, aber zu dem jeweiligen Zeitpunkt auch möglich gewesen wären, berücksichtigt.

Diese Methode setzt voraus, dass Ausführungspfade endlich sind. Das bedeutet insbesondere, dass bei simulierten Eventlogs während der Simulation eine Abbruchbedingung bestimmt werden muss, um potentiell unendlich lange Ausführungspfade zu unterbinden. Um die Unterscheidung zwischen typischen und weniger typischen Pfaden zu berücksichtigen, kann laut den Autoren ein repräsentativer Log gewählt werden. Es stellt sich allerdings die Frage, wie die Repräsentativität eines Logs bestimmt werden kann, insbesondere wenn für die Modelle keine realen Ausführungen zur Verfügung stehen.

Nichtsdestotrotz wird die Methode, mittels Ausführungspfaden einen Ähnlichkeitsabgleich durchzuführen, in Abschnitt 4.5.3.1 für deklarative Prozessmodelle noch einmal aufgegriffen werden. Eine Erweiterung dieser Methode auf M:N-Abbildungen ist nicht direkt möglich, da beispielsweise nicht klar ist, wie eine Menge an Aktivitäten auf einem Pfad gewertet werden soll, die nicht zusammenhängend ist.

2.2.4.3 Abgleich über Ausführungspfadabstraktion

Auch Zha et al. (2010) greifen auf Schaltfolgen in Petri-Netzen, die Ausführungspfaden in Prozessgraphen entsprechen, zurück. Als sogenannte Referenzähnlichkeit definieren die Autoren die Anzahl aller gleichen Schaltfolgen geteilt durch die Anzahl aller unterschiedlichen Schaltfolgen von beiden Prozessmodellen. Da allerdings nicht immer alle Schaltfolgen eines Prozessmodells vollständig bestimmt werden können, führen Zha et al. (2010) die sogenannte TAR-Ähnlichkeit ein. Dazu definieren sie die Nachbarschaftsübergangsrelation (*transition adjacency relation*, TAR), die den Aufbau eines Petri-Netzes voraussetzt: Zwei Transitionen a und b sind in TAR, wenn es eine Spur $\sigma = t_1 t_2 \dots t_n$ aus Transitionen gibt, sodass $t_i = a$ und $t_{i+1} = b$ für $i \in \{1, 2, \dots, n-1\}$. Übertragen auf Prozessgraphen könnte diese Definition lauten, dass zwei Aktivitäten direkt hintereinander ausgeführt werden können, wobei im Prozessmodell nicht unbedingt eine direkte Kante zwischen den beiden Aktivitäten liegen muss; Gateways können sich dazwischen befinden. Die TAR-Ähnlichkeit ist dann der Anteil aller gleichen Paare in den TAR-Mengen der Prozessmodelle geteilt durch die Anzahl aller verschiedener Paare in beiden TAR-Mengen.

Auch dieser Ansatz, komplette Ausführungspfade durch eine Menge an Teilstücken ausgedrückt zu abstrahieren, wird in Abschnitt 4.5.3.2 für deklarative Prozessmodelle noch einmal aufgegriffen. Für eine potentielle Erweiterung auf M:N-Abbildungen gilt das gleiche wie in Abschnitt 2.2.4.2.

2.2.4.4 Abgleich über kausale Fußabdrücke

Der kausale Fußabdruck, der von van Dongen et al. (2006) vorgestellt und von van Dongen et al. (2013) im Rahmen eines Maßes verwendet wird, um das Verhalten von Prozessmodellen zu vergleichen, stellt eine Abstraktion der Ausführungspfade dar. Anstatt alle möglichen Pfade, die durch einen bestimmten Knoten laufen, zu bestimmen und mit den Pfaden des abgebildeten Knotens im anderen Modell zu vergleichen, werden die Vorgänger- und Nachfolgerknoten zu Mengen zusammengefasst. Jeder Knoten erhält so einen Look-Ahead-Link und einen Look-Back-Link. Für die Menge aller Look-Ahead-Links L_{la} gilt, wie von van Dongen et al. (2006) definiert, dass $L_{la} \subseteq (N \times \mathcal{P}(N))$, wobei $(a, B) \in L_{la}$, falls jede Ausführung von a die Ausführung (mindestens) eines $b \in B$ nach sich zieht, d. h., in B sind alle möglichen Nachfolger von a enthalten. Für die Menge aller Look-Back-Links L_{lb} gilt, dass $L_{lb} \subseteq (\mathcal{P}(N) \times N)$, wobei $(A, b) \in L_{lb}$, falls jede Ausführung von b die Ausführung (mindestens) eines $a \in A$ voraussetzt, d. h., in A sind alle möglichen Vorgänger von b enthalten. Der kausale Fußabdruck eines Prozessmodells $G = (N, E, \lambda)$ ist das Tripel $F = (N, L_{lb}, L_{la})$. An dieser Stelle sei darauf hingewiesen, dass diese Definition auch die Bildung von solchen Look-Ahead- bzw. Look-Back-Links erlaubt, die Knoten enthalten, die vom Quellknoten niemals erreicht werden können bzw. von denen der Quellknoten nie erreicht werden kann. Es muss nur ein Knoten in der zugewiesenen Menge sein, der dies erfüllt. Aus der Beschreibung von van Dongen et al. (2006) bzw. van Dongen et al. (2013) geht nicht klar hervor, dass dies wirklich gemeint ist. Die Erklärung zu den Look-Ahead- und Look-Back-Links lässt eher darauf schließen, dass für beispielsweise ein $(a, B) \in L_{la}$ nur solche Knoten in B sein dürfen, die überhaupt von a erreicht werden können und analog in einem $(A, b) \in L_{lb}$ nur solche Knoten in A sein dürfen, von denen aus b erreicht werden kann. Becker und Laue (2012) gehen sogar noch einen Schritt weiter und erlauben für die Look-Ahead- und Look-Back-Links nur minimale Mengen, d. h., dass beispielsweise für ein $(a, B) \in L_{la}$ nur solche Knoten in B sein dürfen, von denen bei einer Ausführung genau einer von a aus erreicht wird, also dass es für $(a, B) \in L_{la}$ kein B' mit $B' \subsetneq B$ geben darf, wofür auch $(a, B') \in L_{la}$ gilt.

Beispiel 2.3. Abbildung 2.15 zeigt ein Prozessmodell, für das einige Beispiele für Look-Ahead- und Look-Back-Links nach den drei genannten Möglichkeiten gezeigt werden.

- Die erste Möglichkeit, wie sie von van Dongen et al. (2006) bzw. von van Dongen et al. (2013) in den Formeln genannt wird, würde für ein $(a, B) \in L_{la}$ z. B. erlauben, dass $(a, B) = (\delta, \{\alpha, \varepsilon\})$, auch wenn α von δ aus gar nicht mehr erreicht werden kann. Da aber ε ein Nachfolgerknoten von δ ist, ist die Definition erfüllt. Auch $(a, B) = (\delta, \{\alpha, \beta, \gamma, \varepsilon\})$, also alle übrigen Knoten außer δ, wäre ein gültiger Look-Ahead-Link.

- Die zweite Möglichkeit, wie sie den Beschreibungen von van Dongen et al. (2006) und van Dongen et al. (2013) nach gedacht war, würde beispielsweise einen Look-Ahead-Link $(a, B) = (\beta, \{\delta, \varepsilon\})$ erlauben, nicht aber einen Look-Ahead-Link $(a, B) = (\beta, \{\varepsilon, \gamma\})$, da γ von β aus nicht erreicht werden kann.

- Die minimale Methode, wie sie von Becker und Laue (2012) vorgeschlagen wird, würde den Look-Ahead-Link $(a, B) = (\beta, \{\delta, \varepsilon\})$ nicht erlauben, da auch $(a, B) = (\beta, \{\delta\})$ ein gültiger Link ist. Ebenfalls ein gültiger Link ist $(a, B) = (\alpha, \{\beta, \gamma\})$, da in einer Ausführung entweder β oder γ von α aus erreicht werden kann, nicht aber beide Knoten.

Die Ähnlichkeitsberechnung zweier kausaler Fußabdrücke erfolgt mit der Kosinus-Ähn-

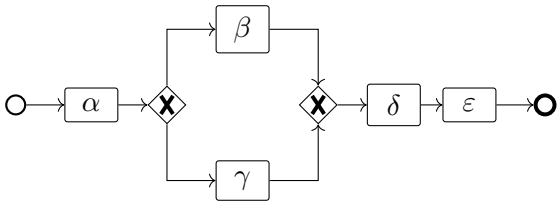


Abbildung 2.15: Beispiel für ein Prozessmodell zur Bestimmung der kausalen Fußabdrücke.

lichkeit aus Abschnitt 2.1, die die Größe des Zwischenwinkels zweier Vektoren im Raum misst und so einen Wert aus $[-1, 1]$ ausgibt. Hierfür müssen zunächst die entsprechenden Vektoren gebildet werden.

In der Menge Θ werden alle Elemente der beiden zu vergleichenden kausalen Fußabdrücke $F_1 = (N_1, L_{1,lb}, L_{1,la})$ und $F_2 = (N_2, L_{2,lb}, L_{2,la})$ vereint: $\Theta = N_1 \cup L_{1,lb} \cup L_{1,la} \cup N_2 \cup L_{2,lb} \cup L_{2,la}$. Die Funktion ι weist jedem Element aus Θ eine laufende Nummer zu: $\iota : \Theta \to \{1, 2, \ldots, |\Theta|\}$. Für die beiden kausalen Fußabdrücke F_1 und F_2 wird dann je ein Fußabdruckvektor φ_1 bzw. φ_2 erstellt. Diese Vektoren sind diejenigen, deren Zwischenwinkel in die Kosinusfunktion eingesetzt wird. Die Einträge $f_{i,j}$ der φ_i, $i = 1, 2$, mit $\varphi_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,|\Phi|})$ sind wie folgt bestimmt:

$$f_{i,\iota(\theta)} = \begin{cases} \frac{1}{2^{|\theta|-1}} & \text{für } \theta \in N_i \cup L_{i,lb} \cup L_{i,la} \\ 0 & \text{sonst,} \end{cases}$$

für $\theta \in \Theta$ mit $|\theta|$ ist die Anzahl der an θ beteiligten Knoten. Für $\theta \in N_i$ ist $|\theta| = 1$ und somit $f_{i,\iota(\theta)} = 1$. Da Knoten mit kleineren Look-Ahead- bzw. Look-Back-Links als informativer angesehen werden, nimmt die Gewichtung der Links mit der Anzahl der beteiligten Aktivitäten exponentiell ab. Je mehr gleiche Knoten in den beiden Prozessmodellen vorkommen, d. h. je größer der Schnitt von N_1 und N_2 und je mehr gleiche Look-Ahead- und Look-Back-Links zwischen den beiden Prozessmodellen bestehen, desto weniger Nullwerte gibt es anteilsmäßig in den beiden Fußabdruckvektoren φ_1 und φ_2. Ein Nullwert in einem Vektor steht immer einem Wert größer Null im anderen Vektor gegenüber, wobei die Größe des Zwischenwinkels von der Größe der Abweichungen in den jeweiligen Koordinaten abhängt. Da alle Einträge in den ϕ_i stets nicht negativ sind, befinden sich die beiden Vektoren für jede Koordinate im selben Halbraum, was dazu führt, dass das Ähnlichkeitsmaß

$$fsim_{cfp}(G_1, G_2) = fsim_{cfp}(F_1, F_2) = \frac{\langle \varphi_1, \varphi_2 \rangle}{\|\varphi_1\|_2 \|\varphi_2\|_2}$$

stets Werte im Intervall $[0, 1]$ liefert. Die Ähnlichkeitsmessung über das Vektormodell ist im Bereich des Information Retrieval eine bewährte Methode. In der ursprünglich vorgeschlagenen Art und Weise der Herleitung der Look-Ahead- und Look-Back-Links wird ein Überfluss an Rechenaufwand produziert, da es für jeden Knoten außer Anfangs- und Schlussknoten sowohl für den Look-Ahead- als auch für den Look-Back-Link eine exponentiell mit der Zahl der Aktivitäten im Modell ansteigende Anzahl an Linkelementen gibt, von denen die meisten über die Gewichtungsfunktion $f_{i,\iota(\theta)}$ nur einen geringen Einfluss auf die Ähnlichkeit haben.

Da das von van Dongen et al. (2013) vorgeschlagene Ähnlichkeitsmaß rein das Verhalten vergleicht, wobei die Aktivitäten über die zugrunde gelegte Abbildung gleichgesetzt werden, schlagen Dijkman et al. (2011) eine Erweiterung der Methode mit Ähnlichkeitswerten der Aktivitätenbeschreibung vor. Es wird also gleichzeitig das Verhalten und die Labelähnlichkeit in

einem Ähnlichkeitsmaß betrachtet. Dazu werden die Gewichte $f_{i,\iota(\theta)}$ mit der Labelähnlichkeit zweier Aktivitäten multipliziert (in der Version von van Dongen et al. (2013) wird hierfür implizit eine Ähnlichkeit von 1 gesetzt), sofern der Quellknoten ein abgebildeter Knoten ist. Mit *sim* als Bezeichung für ein Labelähnlichkeitsmaß ändert sich die Gewichtungsfunktion damit zu

$$f_{i,\iota(\theta)} = \begin{cases} \frac{sim(\alpha,\alpha')}{2^{|\theta|-1}} & \text{für } (\theta = \alpha \in (N_1 \cup N_2) \\ & \quad \vee\ \theta = (\alpha, A) \in L_{la} \vee \theta = (A, \alpha) \in L_{lb}) \\ & \quad \wedge\ (M(\alpha) = \alpha' \vee \alpha = M(\alpha')) \\ 0 & \text{sonst.} \end{cases}$$

Dieses letztgenannte Ähnlichkeitsmaß kombiniert die Verhaltens- und die Labelähnlichkeit, verwendet also mehr Informationen über das Prozessmodell, als Ähnlichkeitsmaße, die nur einen der beiden Aspekte berücksichtigen. Der Nachteil bei diesem Maß ist allerdings, dass durch die Kombination keine separate Gewichtung, wie sie insbesondere in Abschnitt 4.4 eingeführt wird, vorgenommen werden kann. Für die globale Ähnlichkeit aus Definition 2.8, die nur einen Ähnlichkeitsterm $fsim_M$ bei der Berechnung von $gsim_M(G_1, G_2)$ vorsieht, kann diese Kombination jedoch vorteilhaft sein. Für alle vorgestellten Arten der Ähnlichkeitsmessung mittels kausaler Fußabdrücke gilt jedoch wieder, dass nicht klar ist, wie eine Erweiterung auf M:N-Abbildungen erfolgen kann, da schon die Definition der Look-Ahead- und Look-Back-Links nicht (direkt) auf Knotenmengen übertragbar ist. Dagegen ist eine Übertragung der Methoden auf deklarative Prozessmodelle vorstellbar, wenn für diese die Look-Ahead- und Look-Back-Links bestimmt werden können, da für die Ähnlichkeitsmaße selbst keine Informationen zu den Kanten im Prozessmodell benötigt werden. Die Kanten werden allerdings zur Bestimmung der Links verwendet; hier ist für deklarative Prozessmodelle eine andere Methode zu finden.

2.2.4.5 Abgleich über kausale Verhaltensprofile

Weidlich et al. (2010b) stellen in ihrer Arbeit einen Ansatz vor, der das Verhalten eines Prozesses bzw. das im Modell dargestellte Verhalten weiter abstrahiert, was die Berechnung dieses Ähnlichkeitsmaßes effizienter als Bisimulation oder Ausführungspfade macht. Die Definitionen von Weidlich et al. (2010b) beziehen sich auf Prozessmodelle, die als Workflow Nets, eine spezielle Klasse von Petrinetzen, vorliegen, doch sie lassen sich ohne Beschränkung der Allgemeinheit auf Prozessgraphen, wie in Definition 1.1 vorgegeben, übertragen. Zunächst wird die Relation der schwachen Ordnung ($\succ$, *weak order*) eingeführt: Zwei Knoten $(a, b) \in N \times N$ mit $G = (N, E, \lambda)$ sind schwach geordnet, falls b über einen Pfad an Kanten von a aus erreicht werden kann, d. h. $\succ (a, b) \Leftrightarrow \exists(1, \ldots, k) : \{(n_1, n_2), (n_2, n_3), \ldots, (n_{k-1}, n_k)\} \subseteq E \wedge n_1 = a \wedge n_k = b$. Aus dieser Relation werden dann drei weitere Relationen, die das Verhalten zweier Knoten bezüglich ihrer Ordnung näher beschreiben, abgeleitet:

- Strikte Ordnungrelation: $\rightsquigarrow (a, b) \Leftrightarrow a \succ b \wedge b \not\succ a$.
- Ausschließlichkeitsrelation: $+(a, b) \Leftrightarrow a \not\succ b \wedge b \not\succ a$.
- Verschränkte Ordnungsrelation: $||(a, b) \Leftrightarrow a \succ b \wedge b \succ a$.

Für jedes Tupel $(n_1, n_2) \in N$ gilt eine dieser drei Relationen. Die Menge der Relationen über $N \times N$ heißt Verhaltensprofil für N bzw. für das komplette Prozessmodell $G = (N, E, \lambda)$.

Zusätzlich zu diesen ordnungsbeschreibenden Relationen wird der Begriff der Optionalität eingeführt: Ein Knoten ist optional, falls es einen (gültigen) Pfad vom Anfang bis zum Ende des Prozessmodells gibt, ohne dass der Knoten auf diesem Pfad liegt. Aus dem Verhaltensprofil wird durch Einführen einer zusätzlichen Relation, der Kookkurrenzrelation (gemeinsames Vorkommen), das kausale Verhaltensprofil eines Prozessmodells bestimmt:

- Kookkurrenzrelation: $\gg (a, b) \Leftrightarrow (\forall\{(start, n_1), (n_1, n_2), \ldots, (n_k, end)\}$
  mit $\exists i \in \{1, \ldots, k\} : n_i = a \Rightarrow \exists j \in \{1, \ldots, k\} \setminus \{i\} : n_j = b)$

Die Knoten a und b sind dann kookkurrent ($a \gg b$), wenn in jedem Pfad, in dem a auftritt, b ebenfalls auftritt. Ein Knoten n ist damit optional, wenn $start \not\gg n$. Außerdem besteht eine kausale Abhängigkeit zwischen zwei Knoten n_1 und n_2, wenn sie in strikter Ordnungsrelation ($n_1 \rightsquigarrow n_2$) und Kookkurrenzrelation ($n_1 \gg n_2$) sind.

Im Rahmen von Process Mining gibt es auch andere, ähnliche Verhaltensrelationen (siehe z. B. van der Aalst et al., 2004), die jedoch für einen Abgleich bestehender Prozessmodelle in gegebener Form nicht geeignet sind (Weidlich et al., 2010b). Die Abbildung, die für die weitere Berechnung zwischen den beiden zu vergleichenden Prozessmodellen notwendig ist, kann in etwas allgemeinerer Form als in Definition 2.2 gegeben sein. Es sind 1:N-Abbildungen erlaubt, d. h. ein Knoten eines Modells kann mehreren Knoten im anderen Modell entsprechen, wobei ein Knoten in mehreren Abbildungselementen vorkommen darf. Die Notation ist hierbei die folgende: Wird n_1 auf n_2 abgebildet, dann ist $n_1 \sim n_2$ ($n_1 \in N_1, n_2 \in N_2$); es kann, wegen der 1:N-Eigenschaft, aber auch ein $n_3 \in N_2$ geben mit $n_1 \sim n_3$. Ist eine solche Abbildung zwischen den Elementen eines Prozessmodells vorgegeben, können wie in Definition 2.4 die abgebildeten Knoten als Menge $subn_M$ bestimmt werden. Diese Menge kann noch aufgeteilt werden in $subn_{M,1} = subn_M \cap N_1$, die Menge der abgebildeten Knoten des ersten Modells, und $subn_{M,2} = subn_M \setminus subn_{M,1} = subn_M \cap N_2$, die Menge der abgebildeten Knoten des zweiten Modells.

Der Ausdruck $\mathcal{R}_i^\xi$ bezeichne das kausale Verhaltensprofil von Prozessmodell G_i, $i = 1, 2$, mit allen Relationen vom Typ ξ, $\xi \in \{\rightsquigarrow, +, ||, \gg\}$. Ein Knotenpaar $(n, n') \in subn_{M,1} \times subn_{M,1}$ heißt konsistent, falls für alle ξ gilt:

- $\forall m \in subn_{M,2}$ mit $n \sim m$ gilt: $n\mathcal{R}_1^\xi n \Rightarrow m\mathcal{R}_2^\xi m$, falls $n = n'$
  (Falls n auf m abgebildet wird, muss m mit sich selbst in den gleichen Relationen stehen wie n mit sich selbst.)

- $\forall m, m' \in subn_{M,2}$ mit $n \sim m$, $n' \sim m'$ und $m \neq m'$ gilt:

  – $n\mathcal{R}_1^\xi n' \Rightarrow m\mathcal{R}_2^\xi m'$

  oder

  – $n \sim m'$ und $n' \sim m$,

  falls $n \neq n'$
  (Falls n und n' verschieden sind, n auf m und n' auf m' abgebildet wird und auch m und m' verschieden sind, dann muss entweder gelten, dass m und m' in denselben Relationen stehen wie n und n' oder dass n auch auf m' und n' auch auf m abgebildet wird.)

Analog gilt die Definition von konsistenten Knotenpaaren für Knotenpaare $(m, m') \in subn_{M,2} \times subn_{M,2}$. Ein Knotenpaar ist also dann konsistent, wenn das Paar und sein abgebildetes Paar, d. h. dasjenige, das aus jeweils zugeordneten Knoten besteht, entweder in Relation vom selben Typ stehen oder beide Knoten des Paares zu jeweils den beiden Knoten des anderen Modells zugeordnet sind. Die konsistenten Knotenpaare von Modell G_i werden in den Mengen CN_i, $i = 1, 2$ zusammengefasst und das Ähnlichkeitsmaß anschließend über den Anteil der konsistenten Knotenpaare zu allen Knotenpaaren berechnet mittels

$$fsim_{cbp}(G_1, G_2) = \frac{|CN_1| + |CN_2|}{|subn_{M,1} \times subn_{M,1}| + |subn_{M,2} \times subn_{M,2}|} \in [0, 1].$$

Dieser Ansatz der Ähnlichkeitsmessung des Verhaltens von Prozessmodellen über kausale Verhaltensprofile wird später in Abschnitt 4.3.6 noch einmal in angepasster Weise aufgegriffen, da er insbesondere auch auf deklarative Prozessmodelle übertragen werden kann. Des Weiteren wird auch die Eigenschaft der Optionalität in Abschnitt 4.3.3 verwendet. Im Unterschied zu beispielsweise der Methode aus Abschnitt 2.2.4.2, die komplette Ausführungspfade verwendet, ist das Problem potentiell unendlich vieler Pfade nicht relevant, da für die Aussage der schwachen Ordnung nur mindestens ein Pfad gefunden werden muss, der die Ordnung erfüllt. Jedoch müssen alle Pfade gefunden werden, wenn das Fehlen einer schwachen Ordnung gezeigt werden soll. Bei einer Simulation kann jedoch eine endliche Anzahl an Schleifendurchläufen gefunden werden, um jede Kombination von verschiedenen Schleifendurchläufen mindestens einmal zu erhalten. Es kann alternativ auch gezielt nach einzelnen Ordnungen gesucht werden. Eine Erweiterung der Definition von schwacher Ordnung auf M:N-Abbildungen ist in Abschnitt 4.3.5 dargestellt.

2.2.4.6 Abgleich über Verhaltensprofile mit Berücksichtigung unterschiedlicher Flexibilität

Auch von Kunze et al. (2011) werden die Verhaltensprofile (strikte Ordnungsrelation, Ausschließlichkeitsrelation, verschränkte Ordnungsrelation, siehe Abschnitt 2.2.4.5) von Prozessmodellen zum Abgleich herangezogen. Vorausgesetzt wird eine 1:1-Abbildung zwischen den Prozessmodellen. Ein Ähnlichkeitsmaß wird dann separat für jede Relation vom Typ ξ, $\xi \in \{\rightsquigarrow, +, ||\}$, die im Verhaltensprofil $\mathcal{R}_i = \bigcup_\xi \mathcal{R}_i^\xi$ von Prozessmodell G_i, $i = 1, 2$, zusammengefasst sind, mittels des Jaccard-Index berechnet:

$$fsim_\xi(G_1, G_2) = \frac{|\mathcal{R}_1^\xi \cap \mathcal{R}_2^\xi|}{|\mathcal{R}_1^\xi \cup \mathcal{R}_2^\xi|} \in [0, 1]$$

Erweitert wird der Ansatz von Kunze et al. (2011) um ein Ähnlichkeitsmaß, das zusätzlich die Tatsache berücksichtigt, dass eigentlich unabhängige Aktivitäten im Modell in einer beliebigen Reihenfolge modelliert sein können, die jedoch mehr oder weniger willkürlich vom Modellierer gewählt ist. Hierzu wird die umgekehrte, strikte Ordnungsrelation $\rightsquigarrow^{-1}$ benötigt:

- Umgekehrte, strikte Ordnungsrelation: $\rightsquigarrow^{-1}$ $(a, b) \Leftrightarrow \rightsquigarrow (b, a)$

Das Maß $sim_\rightsquigarrow$ wird damit erweitert zu

$$fsim_{\rightsquigarrow'}(G_1, G_2) = \frac{|(\mathcal{R}_1^\rightsquigarrow \cup \mathcal{R}_1^{\rightsquigarrow^{-1}}) \cap (\mathcal{R}_2^\rightsquigarrow \cup \mathcal{R}_2^{\rightsquigarrow^{-1}})|}{|(\mathcal{R}_1^\rightsquigarrow \cup \mathcal{R}_1^{\rightsquigarrow^{-1}}) \cup (\mathcal{R}_2^\rightsquigarrow \cup \mathcal{R}_2^{\rightsquigarrow^{-1}})|} \in [0, 1]$$

Eine ähnliche Überlegung führt zur Erweiterung der Ähnlichkeit über die verschränkte Ordnungsrelation, denn auch hier kann es der Fall sein, dass ein Prozess mit einem unterschiedlichen Grad an Flexibilität modelliert wurde. So erlaubt die verschränkte Ordnung eine flexiblere Ausführung als die strikte Ordnung. Alle Ausführungsmöglichkeiten der strikten Ordnung sind auch in der verschränkten möglich, wobei die verschränkte Ordnung darüber hinaus noch weitere Ausführungspfade erlaubt. Die Erweiterung der verschränkten Ordnungsähnlichkeit lautet wie folgt:

$$fsim_{||'}(G_1, G_2) = \frac{|(\mathcal{R}_1^{\rightsquigarrow} \cup \mathcal{R}_1^{\rightsquigarrow^{-1}} \cup \mathcal{R}_1^{||}) \cap (\mathcal{R}_2^{\rightsquigarrow} \cup \mathcal{R}_2^{\rightsquigarrow^{-1}} \cup \mathcal{R}_2^{||})|}{|(\mathcal{R}_1^{\rightsquigarrow} \cup \mathcal{R}_1^{\rightsquigarrow^{-1}} \cup \mathcal{R}_1^{||}) \cup (\mathcal{R}_2^{\rightsquigarrow} \cup \mathcal{R}_2^{\rightsquigarrow^{-1}} \cup \mathcal{R}_2^{||})|} \in [0, 1]$$

Die Verhaltensähnlichkeit errechnet sich als gewichtetes Mittel über die fünf Einzelähnlichkeiten:

$$fsim_{bp}(G_1, G_2) = \sum_{\xi} w_\xi sim_\xi(G_1, G_2) \in [0, 1],$$

mit $\xi \in \{\rightsquigarrow, +, ||, \rightsquigarrow', ||'\}$, $w_\xi \in [0, 1]$ und $\sum_\xi w_\xi = 1$.

Die verschiedenen Grade an Flexibilität, die bei diesem Ähnlichkeitsmaß berücksichtigt werden, werden in Abschnitt 4.3.6 ebenfalls in angepasster Weise aufgegriffen. Ansonsten sind die Eigenschaften dieser Methode ähnlich zu denen der kausalen Verhaltensprofile aus Abschnitt 2.2.4.5.

Es lässt sich beobachten, dass alle hier aufgeführten Messmethoden zur Verhaltensähnlichkeit bis auf die Methode der Bisimulation direkt oder indirekt auf Ausführungspfade zurückgreifen. Da es oft nicht möglich ist, alle Pfade eines Modells zu betrachten, weil es zum Beispiel unendlich viele sein können, wird versucht, über Mengen (z. B. kausale Fußabdrücke mit Look-Back- und Look-Ahead-Links) oder Relationen (z. B. Verhaltensprofile mit Ordnungsrelationen) zu arbeiten. Es gibt jedoch keine oder zumindest keine prominenten Methoden, die Informationen der Gateways direkt für die Verhaltensähnlichkeit verwenden. Ein Einbeziehen der Gateways kann die Simulation von Pfaden und damit auch die Einschränkungen von Simulationen überflüssig machen. Dies ist beispielsweise in Abschnitt 4.3.6 so durchgeführt.

Alle bisher gezeigten Definitionen von Ähnlichkeit – Labelähnlickeit, Strukturähnlichkeit und Verhaltensähnlichkeit – sind im zweiten Schritt des vierstufigen Ansatzes (Abschnitt 2.2.1.2) verwendbar, da sie auf einer Abbildung der Prozessmodellknoten basieren und die Definition eines Ähnlichkeitsmaßes, wie es in Abschnitt 2.1.1 gegeben ist, erfüllen. Im folgenden Abschnitt werden Abgleichsmethoden vorgestellt, die für den vierstufigen Ansatz zwar nicht verwendbar sind, aber dennoch eine Ähnlichkeit von Prozessmodellen messen.

2.3 Sonstige Methoden zum Modellabgleich

Die Abgleichsmethoden, die in diesem Abschnitt kurz vorgestellt werden, sind Alternativen, die jedoch entweder kein Ähnlichkeitsmaß für den Abgleich verwenden, womit die Ähnlichkeitsberechnung dann nicht Definition 2.1 genügt, Prozessmodelle in nicht grafischer Form voraussetzen oder nicht automatisierbar sind. Dies hat zur Folge, dass beispielsweise keine Entsprechungen von Modellelementen angegeben werden können, womit eine Zusammenlegung von Prozessen schwierig ist, oder dass nicht festgestellt werden kann, ob zwei Modelle

gleich sind (*gleich* im Sinne des verwendeten Maßes, siehe Abschnitt 2.1.2). Die Methoden sind also für den Fortgang der Arbeit nicht weiter relevant, zeigen jedoch, dass der vierstufige Ansatz nur eine Möglichkeit ist, wenn auch eine in der Literatur weit verbreitete, Ähnlichkeit von Prozessmodellen zu bestimmen.

2.3.1 Ähnlichkeit von gewichteten Graphen

Huang et al. (2004) verfolgen einen ähnlichen Ansatz wie Minor et al. (2007), allerdings ist die Funktion, die zum Messen der Ähnlichkeit dient, keine, die die Bedingungen für ein Ähnlichkeitsmaß gemäß Definition 2.1 erfüllt. Huang et al. (2004) betrachten in ihrem Ansatz zwei Prozessmodelle G_1 und G_2 als Graphen mit $G_i = (A_i, E_i, \lambda_i)$, $i = 1, 2$. Für die Ähnlichkeit der Aktivitäten wird ein Ähnlickeitsmaß sim benötigt, das zwei einzelne Aktivitäten miteinander vergleicht, zum Beispiel sim_{sed}. Ursprünglich vergleichen Huang et al. (2004) Prozesse, die (Web-)Services miteinander verknüpfen. Die Funktion sim bezeichnet dann die Ähnlichkeit zweier (Web-)Services und in E sind statt Kanten zwischen Knoten Service-Links enthalten. Die Ähnlichkeit der Web-Services bzw. Aktivitäten ist definiert als

$$sim_{maxA}(A_1, A_2) = \frac{\sum_{j=1}^{|A_1|} \max_{k=1,\ldots,|A_2|} sim(a_{1,j}, a_{2,k}) + \sum_{k=1}^{|A_2|} \max_{j=1,\ldots,|A_1|} sim(a_{1,j}, a_{2,k})}{|A_1| + |A_2|} \in [0, 1]$$

für $a_{i,j} \in A_i$. Für jede Aktivität des einen Modells wird also die nach sim ähnlichste Aktivität des anderen Modells gesucht und über diese Ähnlichhkeitswerte gemittelt. Diese Berechnung entspricht der der Bag of Words-Ähnlichkeit aus Abschnitt 2.2.2.5 mit dem Unterschied, dass hier nicht über (die Ähnlichkeit der) Wörter zweier Aktivitätenbeschriftungen summiert wird sondern über (die Ähnlichkeit der) Aktivitäten zweier Prozessmodelle. Um Kanten miteinander abzugleichen, wird zunächst die Ähnlichkeit zweier einzelner Kanten definiert, und zwar über die mittlere Ähnlichkeit der sie bestimmenden Aktivitäten:

$$sim_e(e_{1,j}, e_{2,k}) = \frac{sim(a_{1,j_s}, a_{2,k_s}) + sim(a_{1,j_e}, a_{2,k_e})}{2}$$

mit $e_{1,j} = (a_{1,j_s}, a_{1,j_e}) \in E_1$ und $e_{2,k} = (a_{2,k_s}, a_{2,k_e}) \in E_2$. Die Variable a_{i,l_s} bezeichnet den Startknoten von Kante $e_{i,l}$ und a_{i,l_e} den Endknoten von Kante $e_{i,l}$, wobei $a_{i,l_s}, a_{i,l_e} \in A_i$ und $e_{i,l} \in E_i$, $i = 1, 2$. Außerdem erhält jede Kante ein Gewicht. Grundsätzlich ist dieses Gewicht 1, außer direkt aus einem XOR-Split ausgehende bzw. in einen XOR-Join eingehende Kanten. Diese erhalten das Gewicht $1/|$ausgehende Kanten des XOR-Splits$|$ bzw. $1/|$eingehende Kanten des XOR-Joins$|$. Wie viele Verzweigungsknoten hierbei verschachtelt sind, spielt keine Rolle, da das Prozessmodell gleichzeitig wie in Abschnitt 2.2.3.4 nach Methode 1 auf Aktivitäten reduziert wird. Ein Beispielmodell in ursprünglicher Form und in auf Aktivitäten reduzierter Form mit Kantengewichten ist in Abbildung 2.16 gegeben. Mit sim_e wird nun die Kantenähnlichkeit sim_{maxE} analog zur Ähnlichkeit zweier Aktivitäten sim_{maxA} so definiert, dass zu jeder Kante aus E_1 die jeweils ähnlichste Kante aus E_2 unter Einbeziehung von sim_e und der Kantengewichte, die mit $w_{e_{\cdot,\cdot}}$ bezeichnet sind, gesucht wird und andersherum zu jeder Kante aus E_2 ebenso. Über diese gewichteten, einzelnen Ähnlichkeitswerte wird

durch Zusammenzählen und Teilen durch die Anzahl aller Kanten gemittelt:

$$sim_{maxE}(E_1, E_2) = \frac{\sum_{j=1}^{|E_1|} \max_{k=1,\ldots,|E_2|} w_{e_{1,j}} w_{e_{2,k}} sim_e(e_{1,j}, e_{2,k}) + \sum_{k=1}^{|E_2|} \max_{j=1,\ldots,|E_1|} w_{e_{2,k}} w_{e_{1,j}} sim_e(e_{1,j}, e_{2,k})}{|E_1| + |E_2|} \in [0,1]$$

Diese beiden Ähnlichkeiten der Aktivitäten und Kanten, sim_{maxA} und sim_{maxE}, werden dann wiederum gewichtet gemittelt:

$$sim_{wg}(G_1, G_2) = \frac{w_1 sim_{maxA}(A_1, A_2) + w_2 sim_{maxE}(E_1, E_2)}{w_1 + w_2} \in [0,1]$$

mit $w_1 + w_2 = 1$ und $w_1, w_2 \geq 0$.

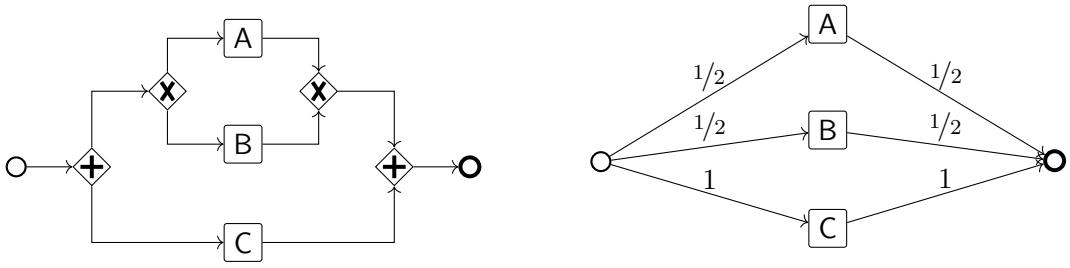


Abbildung 2.16: Beispielmodell in ursprünglicher Form und in auf Aktivitäten reduzierter Form mit Kantengewichten.

Aufgrund der Konstruktion dieses Ähnlichkeitsmaßes, genauer, aufgrund der angegebenen Kantengewichtungen, kann es bei dieser Art der Ähnlichkeitsberechnung vorkommen, dass gleiche Prozessmodelle keinen Ähnlichkeitswert von 1 haben, nämlich dann, wenn optionale Kanten in einem Modell auftauchen. Dies widerspricht der Identitätseigenschaft von Ähnlichkeitsmaßen, wie in Definition 2.1 angegeben, und somit ist sim_{wg} kein Ähnlichkeitsmaß im eigentlichen Sinne. Vor allem erschwert es diese Eigenschaft dem Benutzer, gleiche Modelle, also insbesondere auch Duplikate, zu identifizieren, die bei einer Bereinigung des Modellrepositoriums (Abschnitt 1.1.1) eigentlich überhaupt keinen zusätzlich Aufwand erfordern sollten. Eine relative Aussage über die Ähnlichkeit von Prozessmodellen ist dagegen dennoch möglich.

2.3.2 Ähnlichkeitsabgleich mit Benutzerinteraktion

Auch wenn die meisten Ansätze zum Ähnlichkeitsabgleich als Vorauswahl für Prozessexperten gedacht sind und nicht als absolute Ergebnisse, so gibt es auch einige Ansätze, bei denen explizit das Einschreiten eines Nutzers erforderlich ist. Klinkmüller et al. (2014) verwenden die Bag of Words-Ähnlichkeit (siehe Abschnitt 2.2.2.5) um eine erste Möglichkeit von Modellkorrespondenzen zu erhalten. Der Nutzer lässt sich für ein beliebiges Paar an Prozessmodellen die Ähnlichkeit berechnen. Anschließend trägt er fehlende Korrespondenzen nach und entfernt falsch gesetzte Korrespondenzen. Basierend auf den Falsch-positiv-Raten und Falsch-negativ-Raten für Wortpaare und festgelegten Lernraten wird eine Korrektur der Ähnlichkeitswerte der Wortpaare vorgenommen. Mit Hilfe dieser Korrektur kann ein neuer Ähnlichkeitswert

mittels der Bag of Words-Ähnlichkeit berechnet werden. Dieser Vorgang kann beliebig oft wiederholt werden.

Anstatt einige wenige Prozessexperten nach ihrer Einschätzung zur Ähnlichkeit von Modellen zu befragen, lagern Rodríguez et al. (2016) diese Aufgabe auf eine große Menge an Personen aus. In ihrem Crowdsourcingansatz verwenden sie drei verschiedene Aufgabendesigns: Je zwei Aktivitätsbeschreibungen müssen als ähnlich oder nicht ähnlich eingestuft werden; aus Prozessfragmenten mit drei bis fünf Aktivitäten wird eine Aktivität des ersten Fragments auf eine des zweiten Fragments abgebildet, wobei die Ähnlichkeit des Kontextes dieser beiden Aktivitäten beurteilt werden muss; die Teilnehmer werden mit zwei Prozessfragmenten konfrontiert, wobei sie alle Korrespondenzen, die sie entdecken, selbst eingeben müssen und bis zu zehn Korrespondenzen zwischen zwei Fragmenten erlaubt sind. Zur Validierung des Ansatzes wird die Meinung der Menge mit der Meinung von Experten verglichen. Als Ähnlichkeitsmaß werden hierbei rein subjektive Kriterien verwendet.

Ein großer Nachteil dieser Methode ist, dass sie einerseits einen großen personellen Aufwand bedeutet und somit für paarweise Abgleiche von Modellen in einem großen Repositorium nicht durchführbar ist. Außerdem hängt die Beurteilung der Ähnlichkeit von den Teilnehmern ab und kann bei der Befragung anderer Personen, eventuell sogar bei einer wiederholten Befragung der gleichen Teilnehmer, auch komplett anders ausfallen. Die Wiederholbarkeit der Ergebnisse ist also nicht gegeben. Andererseits können, gerade was die Ähnlichkeit der Aufgabenbeschreibungen angeht, Probleme wie Homonyme, Rechtschreibfehler oder Wortneuschöpfungen umgangen werden. Zum Zweck der besseren Verständlichkeit von Prozessmodellen (Abschnitt 1.1.3) ist der Ansatz nicht geeignet, da der Abgleich gerade deshalb durchgeführt wird, weil eine der Modelliersprachen schwer oder gar nicht verständlich für den/die Benutzer ist.

2.3.3 Abgleich über textuelle Beschreibung

Rana et al. (2016) benutzen anstatt grafischer Prozessmodelle textuelle Beschreibungen von Prozessmodellen, die mit bekannten Textabgleichsmethoden verglichen werden. Zur Überführung von grafischen Modellen, genauer: BPMN-Modellen, wird ein von Leopold et al. (2014) vorgestellter Ansatz verwendet. Die Textabgleichsmethoden, die Rana et al. (2016) verwenden, sind dieselben, die in Abschnitt 2.2.2 für einzelne Aktivitäten bzw. Aufgabenbeschreibungen vorgestellt werden. Der große Nachteil dieser Methode ist, dass Unstimmigkeiten in den verglichenen Prozessmodellen aus dem Abgleichsverfahren nicht direkt ersichtlich sind. Ebenso wenig werden korrespondierende Modellteile angegeben. Außerdem werden zwei Abstraktionsschritte beim Berechnen der Ähnlichkeit vorgenommen, die Transformation des grafischen Modells in eine textuelle Beschreibung und die Abgleichsmethoden auf dieser Beschreibung, was zu einer größeren Verzerrung des Ergebnisses führen kann. Die Methode zur Transformation von Prozessmodellen in textuelle Beschreibungen ist, in der angegebenen Form, nur für BPMN-Modelle gegeben.

2.4 Zusammenfassung der Abgleichsmethoden in verwandten Arbeiten und Einordnung

Die Abgleichsmethoden der verwandten Arbeiten für Prozessmodelle lassen sich zunächst unterscheiden in solche, die ein Ähnlichkeitsmaß, das die Bedingungen aus Definition 2.1 und Gleichung (2.1) erfüllt, zur Berechnung eines Ähnlichkeitswerts verwenden und solche, die

mindestens eine der dort genannten Bedingungen verletzen und somit kein Ähnlichkeitsmaß im eigentlichen Sinne sind. Für die weitere Betrachtung sind nur normierte Ähnlichkeitsmaße von Interesse, um eine Vergleichbarkeit der Methoden und eine Kombinierbarkeit, wie sie im vierstufigen Ansatz von Abschnitt 2.2.1 notwendig ist, zu gewährleisten.

Die normierten Ähnlichkeitsmaße lassen sich dann wiederum einteilen in diejenigen Maße, die eine Abbildung zwischen zwei Prozessmodellen voraussetzen, und in die ohne Abbildung. In der Regel erfolgt eine Abbildung mittels einer partiell injektiven Funktion. Aufgrund der Tatsache, dass bei der Ähnlichkeitsberechnung die korrespondieren Teile der Prozessmodelle und somit insbesondere auch die Unterschiede in den Modellen angezeigt werden sollen, sind auch hier nur die Abgleichsmethoden von weiterem Interesse, die einen Ähnlichkeitswert anhand von Abbildungen errechnen, die wiederum auf Basis des Ähnlichkeitswerts optimiert werden.

Die in der Literatur genannten Ähnlichkeitsmaße greifen vornehmlich auf drei verschiedene Informationen eines Prozessmodells zurück: Aktivitätenbeschreibungen, Prozessverhalten und Modellstruktur. Für den Vergleich von Aktivitätenbeschreibungen wird eine Vielzahl an syntaktischen Vergleichsmethoden (Stringvergleiche) und semantischen bzw. linguistischen Methoden genannt. Durch Kombinationen bewährter Methoden (z. B. Bag of Words-Ähnlichkeit) soll eine Verbesserung der Ergebnisse erziehlt werden. Die strukturbasierten Ähnlichkeitsberechnungen verwenden die Grapheigenschaften eines Prozessmodells nach Definition 1.1. Da bei Betrachtung der reinen Prozessmodellstruktur viele Aspekte eines Prozessmodells ignoriert werden, wird, wie auch im vierstufigen Ansatz, der über die Berücksichtigung des Anteils der abgebildeten Knoten und Kanten Strukturinformation beim Abgleich verwendet, die Ähnlichkeit der Prozessmodellstruktur mit anderen Ähnlichhkeiten, beispielsweise der Labelähnlichkeit, kombiniert. Die Betrachtung des Verhaltens geht über eine reine strukturelle Sichtweise der Prozessmodelle hinaus. Ablaufmöglichkeiten und Eigenschaften des Kontrollflusses wie Nebenläufigkeit oder sich gegenseitig ausschließende Aktivitäten werden berücksichtigt, wobei die Ansätze teils eine starke Abstraktion der Modelle vornehmen. Grundsätzlich sind als Abbildungen nur 1:1-Abbildungen möglich, in seltenen Fällen 1:N-Abbildungen oder eingeschränkte Formen von M:N-Abbildungen, die in Abschnitt 3.3.2 in einer allgemeinen Form definiert werden. Für die bisher gezeigten Methoden ist in Tabelle 2.2 eine qualitative Einschätzung über eine Anwendbarkeit der Methoden und die Qualität der Ergebnisse gegeben, die großteils bereits bei den Methoden direkt aufgeführt sind. Die Anwendbarkeit bezieht sich ausschließlich auf 1:1-Abbildungen, nicht auf M:N-Abbildungen. Die Methoden aus Abschnitt 2.3 sind in der Tabelle nicht aufgeführt.

Alle genannten Methoden setzen imperative Prozessmodelle voraus. Für deklarativ modellierte Prozesse gibt es in der Literatur nur wenige explizit genannte Abgleichsmethoden. In Abschnitt 4.5.1 wird näher darauf eingegangen.

Tabelle 2.2: Qualitative Einschätzung der Abgleichsmethoden in der Literatur.

	Methode	Anwendbarkeit	Qualität
Label	Syntaktische Ähnlichkeit (Levenshtein, Jaro-Winkler, Jaccard, Dice)	Uneingeschränkt und unkompliziert anwendbar, da keine speziellen Voraussetzungen notwendig sind	Bei standardisiertem Vokabular gute Ergebnisse erwartbar; bei freien Beschreibungen keine Möglichkeit, alternative Formulierungen zu erkennen; einzelne Zeichenfehler fallen gering ins Gewicht
	Syntaktische Ähnlichkeit mit Stemming	(wie oben)	bessere Ergebnisse bei verwandten, aber nicht gleichen Formulierungen als synt. Ähnlichkeit; keine Erkennung von nicht wortverwandten Alternativformulierungen
	Semantische Ähnlichkeit	Ontologie/Wörterbuch benötigt, deswegen aufwändig in der Anwendung; nicht für alle Beschriftungen eine klare Berechnungsvorschrift gegeben (mehrere ungleiche Wörter)	Bessere Ergebnisse als bei syntaktischer Ähnlichkeit zu erwarten, da Alternativformulierungen berücksichtigt werden können; Erkennung von Homonymen erfolgt mit einem nicht symmetrischen Maß (kein Ähnlichkeitsmaß)
	Bag of Words	Uneingeschränkt anwendbar; etwas aufwändiger als syntaktische Ähnlichkeit, da ein syntaktisches Ähnlichhkeitsmaß bei der Berechnung benötigt wird	Für standardisiertes Vokabular gut geeignet; sinnentscheidende Wörter können ggf. nicht erkannt werden, d. h. der Ähnlichkeitswert kann dennoch sehr groß sein; Ergebnis hängt auch von verwendetem syntaktischem Ähnlichkeitsmaß ab
	Bag of Words mit Label Pruning	Uneingeschränkt anwendbar; aufwändiger als Bag of Words-Ähnlichkeit, da zusätzlich eine Pruning-Methode angegeben und durchgeführt werden muss	Kein geringer Ähnlichkeitswert durch Vorhandensein unterschiedlich langer Beschriftungen, dadurch allerdings Ignorieren großer Teile der Beschriftung (Zusatzinformation kann so komplett verloren gehen)
	Mittelung über Wortpaarähnlichkeiten	Uneingeschränkt anwendbar; Synonymwörterbuch notwendig	Kein Ähnlichkeitsmaß, da gleiche Beschriftungen nicht Ähnlichkeit 1 haben; dadurch nur relative Aussagen möglich, aber keine weitere Verrechnung mit anderen Ähnlichkeitsmaßen (z. B. Knoten- und Kantenähnlichkeit)
Struktur	Einfache semantische Ähnlichkeit	Unkompliziert anwendbar	Unzureichend, da nur das Vorhandensein von gleichen Knoten den Ähnlichkeitswert hebt; keine Beachtung der Reihenfolge der Aktivitäten
	δ-Vergleichbarkeit	(siehe oben)	(siehe oben)

	Prozessmatrizen	Nicht direkt anwendbar, da nicht als Ähnlichkeitsmaß, sondern als Distanzmaß gegeben	Das Distanzmaß erfüllt die Eigenschaften einer Metrik, insbesondere also der Dreiecksungleichung, was einen Abgleich einer großen Menge an Prozessmodellen beschleunigen kann; inwieweit ein abzuleitendes Ähnlichkeitsmaß dies ebenfalls tut, ist nicht bekannt; Knoten werden nur indirekt über Kanten berücksichtigt
	Anteil gleicher Knoten und Kanten	Grundsätzlich einfach berechenbar, wobei eine Vorverarbeitung der Prozessmodelle erforderlich ist (Entfernung von Gateways)	Durch Gatewayabstraktion können alternative Modellierungen als gleich angesehen werden (keine unnötige Senkung des Ähnlichkeitswerts); Knoten und Kanten werden berücksichtigt; bei Gatewayabstraktion nach Methode 3 können unterschiedliche Modelle evtl. als gleich angesehen werden
	Kontextähnlichkeit mit separater Abbildung	Aufwändig, da eine zusätzliche Abbildung mit Ähnlichkeitsmaß der Beschriftung für jedes Knotenpaar der ursprünglichen Abbildung benötigt wird	Keine Beachtung von Schleifen, die nur einen Knoten umfassen; Ergebnis hängt von zusätzlich benötigtem Labelähnlichkeitsmaß ab
	Kontextähnlickeit ohne separate Abbildung	Einfacher anzuwenden als Kontextähnlichkeit mit separater Abbildung	Abbildung von Gateways grundsätzlich möglich; falls 1:1-Abbildung nur auf Aktivitäten definiert ist, kaum Aussagekraft der Methode, da Aktivitäten stets gleiche Mächtigkeit der Eingangs- und Ausgangskontexte haben
	Merkmalsbasierte Ähnlichkeit	Grundsätzlich für jedes Prozessmodell berechenbar, allerdings eher aufwändig, da viele Fallunterscheidungen getroffen werden; zusätzlich ein Labelabgleich notwendig	Sehr feine Unterscheidung der Knotentypen; bei einer 1:1-Abbildung auf Aktivitäten fallen viele jedoch weg; durch das Ausblenden zu häufiger Rollen, wofür ein Schwellenwert festgelegt werden muss, können wichtige Informationen unbeachtet bleiben; Ergebnis hängt auch vom Labelähnlichkeitsmaß ab
	Prozessmodellblöcke	Auf solche Prozessmodelle anwendbar, die in Blöcke zerlegt und in Baumstruktur transformiert werden können; anschließende Überführung in normalisierten Binärbaum notwendig; nicht explizit als Ähnlichkeitsmaß formuliert, sondern als Distanzmaß	Verschiedene Arten von Verzweigungen werden nicht unterschiedlich behandelt, ansonsten gute Berücksichtigung der Modellstruktur; Ergebnis hängt davon ab, wie aus dem Distanz- ein Ähnlichkeitsmaß gebildet wird

	Kophänetische Distanz	Nicht anwendbar für Modelle mit Aktivitäten auf rückwärts zeigendem Sequenzfluss (bei Schleifen)	Interpretation des resultierenden Wertes nicht klar; es werden vor allem Unterschiede deutlich, nicht aber Korrespondenzen
	(Teil-)Graphisomorphismen/größter, gemeinsamer Teilgraph	Uneingeschränkte Anwendbarkeit	Sehr empfindliche Messung, wobei Grad an Ähnlichkeit (Bildung eines Ähnlichkeitsmaßes) außer bei Isomorphismus nicht klar ist
	Editierabstand	Uneingeschränkt anwendbar; Minimierungsproblem ist zu lösen	Grundsätzlich ein Distanzmaß, aber Transformation zu Ähnlichkeitsmaß möglich; Ergebnis abhängig von festgelegter Höhe der Kosten für unterschiedliche Umformungsoperationen; ist unter bestimmten Kostenfunktionen sogar eine Metrik
Verhalten	Bisimulation	Nur für Transitionssysteme anwendbar	Binäre Aussage: gleich oder ungleich (kein Grad an Ähnlichkeit messbar)
	Komplette Ausführungspfade	Nur sinnvoll anwendbar bei vollständiger oder zumindest repräsentativer Menge an Ausführungspfaden (vollständig: oft nicht möglich; repräsentativ: nicht klar); Simulation/Log notwendig	Falls Voraussetzungen erfüllt sind, sehr genaue Ergebnisse; Erweiterung auf wichtige und unwichtige Pfade möglich
	Ausführungspfadabstraktion	Nur für Transitionssysteme definiert; potentiell unendlich lange Prozessmodelle verursachen keine Probleme	Wegen Abstraktion ungenauere Ergebnisse als bei kompletten Ausführungspfaden zu erwarten
	Kausale Fußabdrücke	Uneingeschränkt anwendbar; relativ aufwändig, da komplette Look-Ahead- und Look-Back-Links bestimmt werden müssen (alternativ kann auch mit einer minimalen Menge der Links gearbeitet werden)	Wegen Abstraktion ungenauere Ergebnisse als bei kompletten Ausführungspfaden zu erwarten
	Kausale Verhaltensprofile	Uneingeschränkt anwendbar; Simulation oder andere Methode zum Finden der schwachen Ordnungen notwendig	Relativ detaillierte Berücksichtigung unterschiedlicher Ausführungsmodalitäten durch unterschiedliche Relationen, jedoch keine Gewichtung von Abweichungen in den verschiedenen Relationen möglich; Kausalität wird mit beachtet

	Verhaltensprofile mit unterschiedlicher Flexibilität	(siehe oben)	Etwas andere Ausführungsmodalitäten als bei kausalen Verhaltensprofilen (keine Kausalität), dafür Berücksichtigung von Flexibilitätsgraden (Enthaltenseinsbeziehung zweier Modelle kann entdeckt werden); Gewichtung der einzelnen Relationen möglich

Kapitel 3

Erweiterungen bisheriger Definitionen

In diesem Abschnitt werden die bisherigen Definitionen von Prozessmodell und Abbildung erweitert. Die aus der Literatur entnommenen Methoden, die in Kapitel 2 vorgestellt werden, verwenden Informationen über die Aufgabenbeschreibungen, die Struktur und das Verhalten von Prozessmodellen. Tatsächlich enthält ein Prozess, und damit auch ein Prozessmodell, aber weitaus mehr Informationen, die ebenfalls für einen Abgleich herangezogen werden können, um dessen Qualität zu steigern. Die Aspekte, die in der Literatur über Ähnlichkeitsabgleiche von Prozessmodellen bislang kaum bis gar nicht berücksichtigt werden und für die somit auch keine Abgleichsmethoden existieren, sind Informationen zu verwendeten Datenobjekten und zum Datenfluss, zu verwendeten Werkzeugen/Services und zu zuständigen Personen/-Gruppen/Rollen. Ein weiteres Problem, das mit den Methoden aus der Literatur nicht zufriedenstellend gelöst werden kann, ist das der unterschiedlichen Granularität bzw. Feinheit von Prozessmodellen. Alle vorgestellten Methoden, die eine Abbildung zwischen Modellelementen voraussetzen, setzen eine 1:1-Abbildung voraus und können, wie bei den einzelnen Methoden aus Kapitel 2 angemerkt, auch nicht auf M:N-Abbildungen erweitert werden. Ein dritter Punkt, der einer Erweiterung bedarf, ist die Tatsache, dass bislang hauptsächlich imperative Prozessmodelle abgeglichen werden. Da deklarative Prozessmodelle zunehmend bekannt werden, ist es sinnvoll, auch für diese Abgleichsmethoden zur Verfügung zu stellen.

In Abschnitt 3.1 werden zunächst noch einmal die beiden Probleme, die die Methoden aus der Literatur aufweisen (keine Beachtung aller Prozessperspektiven, keine Berücksichtigung unterschiedlich feiner Modelle) genauer besprochen. In Abschnitt 3.2 werden dann multiperspektivische Prozessmodelle eingeführt, also solche Modelle, die auch Informationen über Datenobjekte, Services und Agenten bereitstellen. Nachdem in Abschnitt 3.2.1 die Perspektiven zunächst allgemein vorgestellt werden, werden anschließend multiperspektivische imperative Modelle (Abschnitt 3.2.2) definiert und dann, nach einer kurzen Diskussion über Prozessperpektiven, die einzelnen Aktivitäten zugeordnet werden können (Abschnitt 3.2.3), auch multiperpektivische deklarative Prozessmodelle (Abschnitt 3.2.4) eingeführt. In Abschnitt 3.3 wird auf die unterschiedlichen Granularitäten bzw. Abstraktionsgrade von Prozessmodellen eingegangen. Hierfür gibt Abschnitt 3.3.1 eine kurze Übersicht über strukturierte 1:N- und M:N-Abbildungen in der Literatur, ehe in Abschnitt 3.3.2 eine allgemeine M:N-Abbildung definiert wird, die für den weiteren Verlauf der Arbeit relevant ist. Im Anschluss stellt Abschnitt 3.4 eine abstrakte Form des multiperspektivischen Prozessmodells vor, in die sowohl imperative als auch deklarative Modelle überführt werden können. Diese Form wird in dieser Arbeit zur Übertragung von Abgleichsmethoden von imperativen auf deklarative Prozessmodelle benötigt. Die verschiedenen Möglichkeiten der Methodenübertragbarkeit, zum Beispiel

von einer Prozessperspektive auf eine andere, werden in Abschnitt 3.5 vorgestellt. In Abschnitt 3.6 werden Anwendungsfelder von Abgleichsmethoden, das sind bestimmte Kombinationen aus Abbildungsart, Modellart und Prozessperspektive, identifiziert. Für diese Felder wird qualitativ untersucht, wie viele Methoden zur Ähnlichkeitsmessung jeweils aus der Literatur (Kapitel 2) vorhanden sind. Einhergehend mit der Identifizierung der Felder wird außerdem ein Vorgehensmodell beschrieben, wie bei einem Abgleich, insbesondere bei der Auswahl der Anwendungsfelder, verfahren werden kann. Abschnitt 3.7 schließt dieses Kapitel mit einem Überblick über die einzelnen Anwendungsfelder und klärt die Frage, zwischen welchen Feldern Methoden übertragen werden können und für welche Kombinationen zwingend eine Neuentwicklung von Methoden erforderlich ist. Mit der Vorarbeit in diesem Kapitel widmet sich Kapitel 4 anschließend einzig der Erweiterung und Entwicklung von Ähnlichkeitsmaßen für die genannten Anwendungsfelder.

Die Erweiterungen bezüglich der zusätzlichen Prozessperspektiven und der Einführung von M:N-Abbildungen basieren vor allem auf der Arbeit der Autorin und der entsprechenden Co-Autoren (Baumann et al., 2014). Zusätzlich stellt die Arbeit von Baumann et al. (2016a) die Grundlage für den Abgleich von deklarativen Prozessmodellen dar.

3.1 Einschränkungen bisheriger Ansätze

Die bislang vorgestellten Ansätze aus der Literatur offenbaren vor allem in zwei Bereichen Mängel, die im Folgenden näher erläutert werden sollen. Diese Probleme werden in der Literatur ebenfalls adressiert (Grigori et al., 2010) und als noch (teilweise) offen bezeichnet.

Ein großes Problem der bisherigen Ansätze liegt darin, dass die Abgleichsmethoden auf einer Definition von Prozessmodellen basieren, die nur die funktionale Perspektive, das heißt die Aktivitätenbeschreibung, die grafische Struktur und die verhaltensorientierte Perspektive, also den Kontrollfluss, über die Kanten berücksichtigt. Ein Prozess besteht aber aus weiteren Perspektiven, wie beispielsweise den involvierten Agenten oder übermittelten Datenobjekten. Durch ein Einbeziehen der bislang nicht betrachteten Perspektiven in die Ähnlichkeitsmessung können genauere Ergebnisse erzielt werden, wobei die Relevanz der einzelnen Perspektiven individuell für jeden Abgleich oder jede Modellsammlung festgelegt werden kann. Außerdem ist die Ähnlichkeitsmessung durch die Betrachtung mehrerer Perspektiven robuster gegenüber Fehlern in einzelnen Perspektiven.

Ein weiteres Problem der bisherigen Ansätze besteht bei der Identifikation korrespondierender Prozessmodellelemente. Bei Prozessmodellen mit unterschiedlicher Granularität, also mit unterschiedlichen Abtraktionsebenen, kann eine 1:1-Abbildung gemäß Definition 2.2 keine allzu guten Ergebnisse erwarten lassen, da zwangsläufig viele Knoten gelöscht werden müssen (Castelo Branco et al., 2012a). Es können nur maximal so viele abgebildet werden, wie das kleinere Modell Knoten hat, siehe zum Beispiel Abbildung 3.1. Im Modell auf der rechten Seite (G_2) müssen immer mindestens drei Knoten gelöscht werden, weil nicht mehr 1:1-Korrespondenzen aufgrund der Knotenanzahl des linken Modells (G_1) möglich sind. Dies führt, unabhängig von der Abbildung, zu einem kleinen Wert von $fsubn_M$. Zwar kann die Gewichtung von $fsubn_M$ bei der Berechnung der Prozessmodellähnlichkeit nach Definition 2.8 herabgesetzt werden, jedoch werden dann Abbildungen, die grundsätzlich nur wenige Korrespondenzen anbieten, nicht genügend bestraft. Folglich ist, um dieses Problem der Granularitätendifferenz zu beheben, eine Erweiterung der 1:1-Abbildung auf eine Abbildung wünschenswert, die es erlaubt, mehrere Knoten auf einen Knoten abzubilden oder gar mehrere Knoten auf mehrere Knoten. Dies stellt die Motivation für die nächsten beiden Abschnitte 3.2

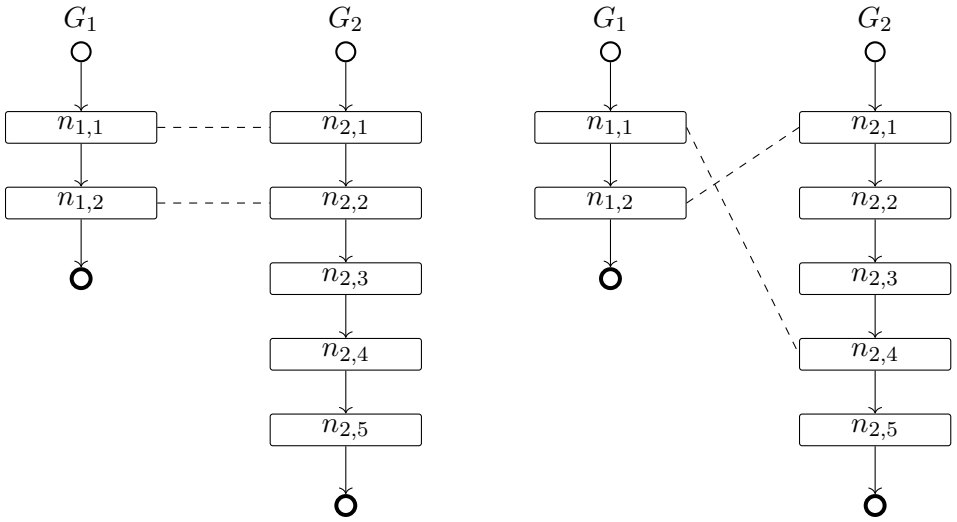


Abbildung 3.1: Zwei Möglichkeiten für eine 1:1-Abbildung zwischen zwei Prozessmodellen G_1 und G_2 mit stark unterschiedlicher Granularität; die Abbildung ist mittels getrichelter Linien angedeutet.

und 3.3 dar, zum einen multiperspektivische Prozessmodelle einzuführen und zum anderen die möglichen Abbildungen zwischen Prozessmodellen auszuweiten. Das Hinzunehmen von deklarativen Prozessmodellen ist eine Maßnahme, die über die zwei genannten Punkte hinausgeht. Dass durch diese Erweiterungen weitere Probleme impliziert werden, sollte nicht überraschen. Es ist nicht klar, wie für multiperspektivische Prozessmodelle, die mittels einer M:N-Abbildung miteinander verglichen werden, oder für deklarative Prozessmodelle ein Ähnlichkeitswert bestimmt werden kann. Dieser Fragestellung wird in Kapitel 4 nachgegangen.

3.2 Multiperspektivische Prozessmodelle

Wie schon von Yan et al. (2010) angesprochen, können für eine Ähnlichkeitsbestimmung auch weitere Aspekte eines Prozessmodells, abgesehen von Beschriftung, Verhalten und Struktur, verwendet werden, sofern sie in den zu vergleichenden Modellen gegeben sind. Mehr betrachtete Aspekte erhöhen zwar die Laufzeit der Berechnung, es kann jedoch ebenfalls davon ausgegangen werden, dass sich die Qualität der Ergebnisse verbessert.

Da ein Prozess mindestens fünf verschiedene Perspektiven abdeckt, nämlich die Aktivitätenbeschreibung (*funktionale Perspektive*), die involvierten Agenten (*organisatorische Perspektive*), die eingehenden und ausgehenden Datenobjekte (*Datenperspektive*), die verwendeten Services (*operationale Perspektive*) und die Flussabhängigkeiten der verschiedenen Aufgaben (*verhaltensorientierte Perspektive*) (Jablonski und Bussler, 1996), sollte auch ein Prozessmodell diese Perspektiven widerspiegeln, um als geeignetes Instrument für die Dokumentation oder die Ausführung eines Prozesses angesehen werden zu können.

Eine besondere Form der multiperspektivischen Prozessmodelle sind perspektivenübergreifende Prozessmodelle (Zeising et al., 2014; Jablonski und Bussler, 1996). Darunter versteht man Prozessmodelle, die zum einen die verschiedenen Aspekte eines Prozesses abbilden und zum anderen in ihren modelldefinierenden Regeln eine beinahe beliebige Komplexität erlauben, also zum Beispiel auch Regeln beinhalten, die drei oder mehr verschiedene Perspektiven

miteinander verknüpfen. Auch perspektivenübergreifende Prozessmodelle werden im Verlauf dieser Arbeit angesprochen (vgl. Abschnitt 3.2.4).

3.2.1 Die fünf Perspektiven eines Prozesses

Zur Darstellung eines Prozesses werden von Curtis et al. (1992) zunächst vier Perspektiven genannt:

- Die funktionale Perspektive gibt an, was während des Prozesses durchgeführt wird, also welche Aufgaben erledigt werden.
- Die verhaltensorientierte Perspektive gibt an, wann und wie die Aufgaben durchgeführt werden. Es kann die Reihenfolge eine Rolle spielen aber auch bestimmte Modalitäten wie Unabhängigkeit oder Ausschließlichkeit von Aufgaben.
- Die organisatorische Perspektive gibt an, von wem welche Aufgabe ausgeführt wird bzw. werden kann.
- Die informationale Perspektive, auch datenorientierte Perspektive oder kurz Datenperspektive genannt, gibt an, welche informationellen Einheiten bei welchen Aufgaben benötigt, z. B. erzeugt oder manipuliert, werden.

Jablonski und Bussler (1996) erweitern die genannten Perspektiven um die operationale Perspektive:

- Die operationale Perspektive gibt an, welche Werkzeuge und Services zur Durchführung einer Aufgabe benötigt werden.

Von van der Aalst et al. (2003b) werden diese fünf Perspektive aufgegriffen und die gesamte Geschäftsprozessverwaltung (*Business Process Management*, BPM) davon abhängig gemacht: „Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information.“ Von Schönig et al. (2014) wird auch eine

- ortsbezogene Perspektive genannt, die den Aufenthaltsort von Mitarbeitern und die Orte, an denen Aufgaben erledigt werden müssen, im Modell widerspiegelt. Es wäre auch denkbar, eine
- Zeitperspektive im Sinne von Datums- und Uhrzeitangaben im Modell festzuhalten.

Da für die beiden letztgenannten Perspektiven im Moment jedoch noch keine verbreitete Modellierung zur Verfügung steht, werden sie im Folgenden vernachlässigt und sich auf die erstgenannten fünf Prozessperspektiven beschränkt. Eine Erweiterung der Abgleichsmethoden um weitere Perspektiven ist jedoch aufgrund des modularen Aufbaus des in dieser Arbeit entwickelten Ähnlichkeitsmaßes jederzeit möglich.

Die fünf betrachteten Perspektiven sind unabhängig von der verwendeten Modelliersprache, also insbesondere auch unabhängig vom gewählten Modellierparadigma, sowie untereinander unabhängig (Jablonski und Bussler, 1996). Nicht immer sind in einem Prozessmodell alle Perspektiven vertreten, da nicht jede Modelliersprache alle Perspektiven abdeckt. In einem

Petri-Netz sind beispielsweise lediglich die funktionale und die verhaltensorientierte Perspektive zu finden, jedoch werden in diversen Erweiterungen auch andere Perspektiven abgebildet, z. B. im Coloured-Petri-Net die datenorientierte Perspektive (Jensen, 2013). In grafischen BPMN-Modellen sind die funktionale, die verhaltensorientierte, die organisatorische (über die Zuordnung von Aktivitäten zu Pools und Lanes) und die datenorientierte Perspektive (assoziierte Datenobjekte und Datenspeicher) darstellbar. In der deklarativen Modelliersprache DECLARE bzw. ihrer graphischen Repräsentierung ConDec sind ebenfalls nur die funktionale und die verhaltensorientierte Perspektive sichtbar. Auch hier gibt es Erweiterungen, um die übrigen Perspektiven mit einzubeziehen. So wird von Montali et al. (2013) eine Erweiterung von DECLARE um die datenorientierte Perspektive vorgeschlagen. In der textuellen Declarative Process Intermediate Language (DPIL) ist es möglich, alle fünf Perspektiven eines Prozesses zu modellieren und perspektivenübergreifende Regeln zu formulieren (Zeising et al., 2014). Regeln in DPIL sind in Prädikatenlogik erster Ordnung formuliert, woraus ihre komplexe Aussagekraft folgt. Durch Definitionen von Makros können bestimmte Regelausdrücke jedoch wiederverwendet werden.

3.2.2 Erweiterung der imperativen Prozessmodelldefinition

Um die verschiedenen Perspektiven eines Prozesses bei der Modellierung zu berücksichtigen, muss die Definition von Prozessmodell aus Abschnitt 1.3.2 um eben diese Perspektiven erweitert werden. Diejenigen Perspektiven, die im Vergleich zu Definition 1.1 neu hinzukommen, sind die organisatorische, die operationale und die datenorientierte Perspektive. Jedem Knoten bzw. jeder Aktivität kann je eine Menge an Agenten, Werkzeugen und Datenobjekten zugeordnet werden, wobei bei den Datenobjekten auch feiner unterschieden werden kann in eingehende und ausgehende Objekte (La Rosa et al., 2011) oder sogar in konsumierte, verarbeitete und produzierte Objekte.

Die folgende Definition erweitert Definition 1.1 des imperativen Prozessmodells um die eben genannten Perspektiven. Um die Notation kompakter zu halten, wird nicht zwischen eingehenden und ausgehenden Datenobjekten unterschieden, es werden vielmehr verwendete Datenobjekte betrachtet. Die Unterscheidung in eingehende und ausgehende kann jedoch problemlos durchgefürt werden. Anstatt einer Menge an Datenobjekten, die jeder Aktivität zugeordnet ist, sind es bei einer Unterscheidung zwei Mengen, die in den Ähnlichkeitsberechnungen jeweils komplett analog zur Menge der verwendeten Datenobjekte behandelt werden können.

Definition 3.1 (Multiperspektivisches, imperatives Prozessmodell)**.** Ein multiperspektivisches Prozessmodell G ist ein Tupel der Form $G = (N, E, \lambda)$. Es sei N eine Menge an Knoten, die sich disjunkt aus Aktivitäten A, Verzweigungsknoten C sowie einem Startevent e_{start} und einem Endevent e_{end} zusammensetzt; $N = A \dot\cup C \dot\cup \{e_{start}, e_{end}\}$ mit $C = XOR_s \dot\cup XOR_j \dot\cup AND_s \dot\cup AND_j$. Die Menge $E \subseteq N \times N$ ist eine Menge an Kanten, für die gilt:

- $\forall n \in N \setminus A : (n, n) \notin E$
- $|\{(e_{start}, n) \in E\}| = 1$, $|\{(n, e_{start}) \in E\}| = 0$
- $|\{(n, e_{end}) \in E\}| = 1$, $|\{(e_{end}, n) \in E\}| = 0$
- $\forall a \in A : |\{(a, n) \in E | n \in N \setminus \{a\}\}| = 1 \wedge |\{(n, a) \in E | n \in N \setminus \{a\}\}| = 1$

- $\forall c_s \in \{XOR_s, AND_s\} : |\{(n, c_s) \in E | n \in N\}| = 1 \wedge |\{(c_s, n) \in E | n \in N\}| > 1$
- $\forall c_j \in \{XOR_j, AND_j\} : |\{(n, c_j) \in E | n \in N\}| > 1 \wedge |\{(c_j, n) \in E | n \in N\}| = 1$

Das Startevent hat genau eine ausgehende Kante, das Endevent genau eine eingehende. Jede Aktivität hat jeweils genau eine ein- und eine ausgehende Kante von bzw. zu einem Knoten, der nicht selbst die Aktivität ist. Jedes Split-Gateway hat eine eingehende und mindestens zwei ausgehende Kanten und jedes Join-Gateway mindestens zwei eingehende und eine ausgehende Kante. Kanten von Knoten auf sich selbst gibt es nur für Aktivitäten, was als Kurzschreibweise für Loops, die nur aus einer Aktivität bestehen, aufgefasst werden kann.

Weiter seien $\mathcal{L}$ eine Menge an Beschriftungen, $\mathcal{A}$ eine Menge an Agenten bzw. Rollen, $\mathcal{D}$ eine Menge an Dokumenten und $\mathcal{S}$ eine Menge an Werkzeugen und Services. Die Funktion λ ordnet jeder Aktivität eine Beschriftung, eine Menge an Agenten/Rollen, eine Menge an Dokumenten sowie eine Menge an Werkzeugen/Services zu:

$$\lambda : A \to \mathcal{L} \times \mathcal{P}(\mathcal{A}) \times \mathcal{P}(\mathcal{D}) \times \mathcal{P}(\mathcal{S}),$$

also $\lambda(a) = (\lambda_1(a), \lambda_2(a), \lambda_3(a), \lambda_4(a))$ mit $\lambda_1(a) \in \mathcal{L}$, $\lambda_2(a) \subseteq \mathcal{A}$, $\lambda_3(a) \subseteq \mathcal{D}$ und $\lambda_4(a) \subseteq \mathcal{S}$.

Zur Zuordnung von Agenten-, Datenobjekt- und Werkzeugmengen zu Aktivitäten sei angemerkt, dass für $\lambda_2(a) = \{A_1, A_2, \ldots, A_k\}$ gilt, dass Aktivität a von den Agenten A_1 oder A_2 oder ... oder A_k ausgeführt werden kann, d. h. alle genannten Agenten sind möglich, aber die Ausführung erfolgt dann tatsächlich nur von einem der genannten Agenten bzw. von einer Person, die eine der genannten Rollen erfüllt. Für $\lambda_3(a) = \{d_1, d_2, \ldots, d_l\}$ und $\lambda_4(a) = \{s_1, s_2, \ldots, s_r\}$ gilt, dass Aktivität a alle genannten Dokumente und Werkzeuge benötigt, bei der Ausführung von Aktivität a werden d_1 und d_2 und ... und d_k sowie s_1 und s_2 und ... und s_r benötigt. Insgesamt muss ein nach Definition 3.1 gebildetes Prozessmodell fehlerfrei (*sound*) sein, um einen korrekten Ablauf des Prozesses gemäß des Modells zu gewährleisten (Polyvyanyy et al., 2009), d. h. es dürfen beispielsweise keine Sackgassen (*deadlocks*) oder Teufelskreise (*traps, vicious circles*) enthalten sein (van Dongen et al., 2006).

3.2.3 Perspektiven als Eigenschaften von Aktivitäten

Wie in Abschnitt 3.2.2 geschrieben, wird jeder Aktivität je eine Menge an Agenten, Werkzeugen und Datenobjekten zugeordnet, wobei diese Zuordnung natürlich von der verwendeten Modelliersprache abhängt. In BPMN werden Agenten über die Einsortierung der Aktivitäten in Pools bzw. Swimlanes zugeordnet, Datenobjekte werden als Artefakte direkt mittels einer gestrichelten Linie den Aktivitäten zugewiesen. Auch für DECLARE gibt es Ansätze, weitere Perspektiven als nur die funktionale und die verhaltensorientierte abzubilden. So führen Montali et al. (2013) die datenorientierte Perspektive in DECLARE ein, wobei auch hier die Datenobjekte als eingehende oder ausgehende Daten an Aktivitäten angebunden werden. Für die deklarative Sprache DPIL stellen Zeising et al. (2014) einige Makros vor, die Agenten und Datenobjekte in ähnlicher Weise an Aktivitäten knüpfen. Somit bilden in der Regel die Aktivitäten das Grundgerüst eines Prozessmodells und Agenten, Werkzeuge und Datenobjekte können, genau wie die Aktivitätenbeschreibung, als Eigenschaften bzw. Merkmale von Aktivitäten aufgefasst werden, die zunächst einmal von der Verknüpfung der Aktivitäten, also auch dem Verhalten des Prozessmodells, unabhängig sind. Für deklarative Prozessmodelle folgt damit, dass die Definition dieser Modelle analog zu der imperativer Prozessmodelle um die noch fehlenden Perspektiven erweitert werden kann.

Sollen mehr Perspektiven betrachtet werden als die in Definition 3.1 angegebenen, so kann die Definition einfach über eine neue Komponente der Funktion λ erweitert werden, wenn die Perspektive als Aktivitätseigenschaft aufgefasst werden kann. So können beispielsweise die verwendeten Datenobjekte einfach in eingehende und ausgehende Datenobjekte aufgeteilt und getrennt betrachtet werden, wie vor Definition 3.1 erwähnt.

3.2.4 Erweiterung der deklarativen Prozessmodelldefinition

Die Definition deklarativer Prozessmodelle (Definition 1.2) wird analog zur Definition multiperspektivischer, imperativer Prozessmodelle (Definition 3.1) um die noch fehlenden Perspektiven erweitert. Die Zuordnung der verschiedenen Ressourcen erfolgt nicht über eine statische Funktion λ wie für imperative Modelle, sondern wird über die Regelmenge $\mathcal{C}$ erreicht, deren Regeln nun auch auf die Mengen der Agenten, der verwendeten Datenobjekte und der Werkzeuge definiert werden können.

Definition 3.2 (Multiperspektivisches, deklaratives Prozessmodell)**.** Ein multiperspektivisches, deklaratives Prozessmodell $S = (A, \mathcal{L}, \mathcal{A}, \mathcal{D}, \mathcal{S}, \ell, \mathcal{C})$ besteht aus einer endlichen Menge an Aktivitäten A und deren Beschreibungen $\mathcal{L}$, einer Menge $\mathcal{A}$ an Agenten/Rollen, einer Menge $\mathcal{D}$ an Dokumenten, einer Menge $\mathcal{S}$ an Services und einer endlichen Menge an Regeln C, welche die Randbedingungen der Prozessausführung vorgeben. Den Aktivitäten wird über die Funktion $\ell : A \to \mathcal{L}$ direkt jeweils eine Beschreibung zugewiesen.

Der Begriff des perspektivenübergreifenden Prozessmodells ist bislang vor allem im Zusammenhang mit deklarativen Prozessmodellen von Bedeutung, wobei dies weniger mit dem Modellierparadigma, als vielmehr mit den verwendeten Sprachen zu tun hat. Als perspektivenübergreifend wird dabei eine Regel bezeichnet, die Aktivitäten nicht nur mit einer der genannten Ressourcen, sondern mit mehreren verknüpft (Zeising et al., 2014). Beispielsweise ist dies bei der Regel „Wenn der Überweisungsauftrag geringer als 500 EUR umfasst, ist keine zusätzliche Kontrolle erforderlich (maschinelle Ausführung); überschreitet er diesen Wert, so muss ein qualifizierter Mitarbeiter den Auftrag nach einer Kontrolle von Hand bestätigen“ der Fall. Der Ausführende ist hierbei abhängig vom Wert eines bestimmten Datenobjekts, der erst zur Laufzeit, d. h. während der Ausführung des Prozesses, bestimmt wird. Da allerdings der Ähnlichkeitsabgleich von Prozessmodellen, wenn er nicht die Ausführungspfade verwendet, auf den Modellen unabhängig von einer Ausführung durchgeführt wird, können solche Regeln, abhängig von der angewendeten Abgleichsmethode, nur bedingt berücksichtigt werden. Es kann jedoch jeder Menge an Aktivitäten, egal ob in einem imperativen oder deklarativen Prozessmodell, schon zum Zeitpunkt der Modellierung eine maximal mögliche Menge an Ressourcen zugeordnet werden. Das bedeutet, dass auch wenn zum Modellierzeitpunkt noch nicht klar ist, welche konkreten Prozessinstanzen auftreten werden, und auch wenn möglicherweise klar ist, dass bestimmte Ressourcen sich gegenseitig ausschließen, trotzdem zu jeder Aktivität eine größtmögliche Menge an Agenten, an Datenobjekten und an Services angegeben werden kann, da die Mengen aller für den Prozess zur Verfügung stehenden Ressourcen beschränkt sind. Dies erlaubt es, auch perspektivenübergreifende Prozessmodelle, deren Einschränkungen erst während der Ausführung konkretisiert werden, bereits zum Modellierzeitpunkt mit anderen Modellen oder untereinander zu vergleichen. Definition 3.2 kann auch als Definition für perspektivenübergreifende, deklarative Prozessmodelle dienen, wenn die Regeln entsprechende Formulierungen zulassen.

3.3 Prozesse mit unterschiedlichen Abstraktionsgraden

Unter anderem von Castelo Branco et al. (2012b) wird genannt, dass auch Prozessmodelle mit einem unterschiedlichen Grad an Abstraktion, also einer unterschiedlichen Granularitätsstufe, in der Praxis miteinander verglichen werden müssen. Hierbei ist es nicht notwendigerweise der Fall, dass die Verfeinerung hierarchisch besteht, ein Prozessmodell also eine Verfeinerung eines anderen ist (Weidlich et al., 2009). Die Frage nach der Granularität eines Prozessmodells stellt sich bereits zum Zeitpunkt der Modellierung, wobei es hierfür keine allgemeingültige Antwort gibt, wie feingranular bzw. wie grob ein Prozess modelliert sein soll (Curtis et al., 1992). Unter anderem kann oder sollte sogar, wie Curtis et al. (1992) schreiben, die Granularität an die Zielgruppe des Modells und deren Kenntnisstand angepasst sein.

Falls für zwei Prozessmodelle $G_1 = (N_1, E_1, \lambda_1)$ gilt, dass $|N_1| < |N_2|$, dann können auf keinen Fall alle Knoten aus N_2 über eine Abbildung M gemäß Definition 2.2 abgebildet werden. Wird der Abgleichsansatz aus Abschnitt 2.2.1 verwendet, bei dem sich nicht abgebildete Knoten negativ auf die Ähnlichkeit auswirken, so kann die Ähnlichkeit von vornherein einen bestimmten Schwellenwert, der von der Anzahl der Aktivitäten beider Modelle abhängt, nicht überschreiten. Um diesem Problem, das über die Definition der Abbildung, die ja eine 1:1-Abbildung ist, entsteht, zu begegnen, wird bereits von Dijkman et al. (2009b) eine Erweiterung der Abgleichsmethode vorgeschlagen. Nachdem eine 1:1-Abbildung zwischen den zu vergleichenden Prozessmodellen gebildet wurde, wird nach und nach für die übrig gebliebenen Knoten geprüft, ob sie zu einem der bestehenden Abbildungselemente so hinzugefügt werden können, dass sich der Ähnlichkeitswert durch dieses Hinzunehmen verbessert. Das Problem, dass zu Beginn lediglich eine 1:1-Abbildung angenommen werden darf, besteht allerdings weiterhin. Wird das anfängliche Überschreiten von Schwellenwerten für das Erkennen einer Ähnlichhkeit verlangt, können eigentlich bestehende Ähnlichkeiten zwischen Mengen an Aktivitäten mit der Methode von Dijkman et al. (2009b) dann nicht erkannt werden, wenn der Ähnlichkeitswert keines der einzelnen Paare den Schwellenwert überschreitet. Von Baumann et al. (2014) ist für diesen Fall ein Beispiel gegeben. Außerdem erklären Dijkman et al. (2009b) nicht, wie ein Ähnlichkeitswert nach dem Hinzunehmen eines Knotens zu einem bestehenden Paar berechnet wird. Es ist also notwendig, von vornherein Mengen an Aktivitäten bei einem Abgleich zu berücksichtigen und ein Ähnlichkeitsmaß direkt auf Mengen zu definieren.

In Abschnitt 3.3.1 werden weitere Ansätze aus der Literatur genannt, die über 1:1-Abbildungen hinausgehen, um dem Granularitätenproblem zu begegnen. Diese unterliegen jedoch individuellen Einschränkungen, sodass in Abschnitt 3.3.2 beliebige M:N-Abbildungen eingeführt werden.

3.3.1 Strukturierte 1:N- und M:N-Abbildungen

Die Arbeit von Weidlich et al. (2010a) führt 1:N-Abbildungen ein, d. h., es wird berücksichtigt, dass jeweils eine Aktivität in einem Prozessmodell durch mehrere Aktivitäten in einem zweiten Prozessmodell ausgedrückt werden kann. Dies erlaubt, die von Castelo Branco et al. (2012b) genannten hierarchischen Verfeinerungen (jede Aktivität des einen Modells auf 0 bis N des anderen abzubilden) zu berücksichtigen, nicht jedoch die ebenfalls genannten nicht hierarchischen (jeweils eine Aktivität des einen Modells auf 0 bis N des anderen Modells oder 0 bis N des einen Modells auf eine Aktivität des anderen Modells abzubilden). Auch Mischformen, das sind Fälle, bei denen beispielsweise zwei Aktivitäten in einem Modell dreien im zweiten Modell entsprechen, können so nicht berücksichtigt werden. Schon im 1:N-Fall, der von Weidlich et al. (2010a) beschrieben wird, ist die Anzahl an möglichen Matches, die

gebildet werden können, sehr groß, da diese Anzahl hyperexponentiell von der Größe der Prozessmodelle abhängt. Um dieser kombinatorischen Explosion Herr zu werden, schränken Weidlich et al. (2010a) die möglichen mehrelementigen Mengen auf solche ein, die bestimmte Nachbarschaftsbedingungen, wie sequentielle Abfolgen, Verzweigungen oder Zusammenflüsse, die innerhalb einer vorgegebenen Distanz liegen, erfüllen.

Auch Castelo Branco et al. (2012a) berücksichtigen in ihrer Arbeit Prozessmodelle unterschiedlicher Granularität. Um auf diesen einen Abgleich durchzuführen, wird zunächst der Begriff der single-entry single-exit (SESE)-Region eingeführt (Johnson et al., 1994). Eine SESE-Region ist hierbei jeder Subgraph, der genau eine eingehende Kante und genau eine ausgehende Kante hat. Jede einzelne Aktivität ist beispielsweise eine SESE, aber auch das komplette Prozessmodell bei eindeutigem Start- und Endknoten. Anhand der SESE-Regionen lässt sich ein Prozessmodell dann in einen eindeutigen Prozessstrukturbaum (*(Refined) Process Structure Tree*, (R)PST) umformen (Vanhatalo et al., 2008; Polyvyanyy et al., 2011). Der Abgleich wird dann auf den (R)PSTs der zu vergleichenden Prozessmodelle durchgeführt. Korrespondenzen können zwischen zwei einzelnen Prozessmodellelemente, die im (R)PST in den Blättern repräsentiert sind, zwischen einem einzelnen Element und einer SESE-Region (mit mehr als einem Element) oder zwischen zwei SESE-Regionen (mit jeweils mehr als einem Element) vorliegen. Der letzte Fall bildet also eine Menge von Modellelementen des ersten Modells auf eine Menge von Modellelementen des zweiten Modells ab. Diese Mengen können jedoch immer nur vollständige SESE-Regionen sein.

Von Polyvyanyy et al. (2012) werden ebenfalls M:N-Abbildungen zwischen Prozessmodellen erlaubt. Einzelne Prozessmodellknoten können hierbei mehrfach in der Abbildung auftauchen, das heißt, die Abbildung ist überlappend. Es werden allerdings Bedingungen an die erlaubten Abbildungen gestellt: Aus den Knoten eines Modells dürfen nur solche (überlappenden) Knotenmengen gebildet werden, sodass für alle einzelnen Knoten aus je zwei Mengen dieselbe Kausalität (Reihenfolge) gilt. Auch Abbildungen müssen diese Reihenfolge einhalten, d. h., für die Mengen im Bild der Abbildung muss dieselbe Reihenfolge gelten. Wie eine solche Abbildung gefunden wird, wird von Polyvyanyy et al. (2012) nicht spezifiziert. Eine semantische Ähnlichkeit, welcher Art auch immer, wird zwischen den Knoten vorausgesetzt. Alle angegebenen Definitionen beruhen dabei auf Petrinetzen. Eine Ähnlichkeitsberechnung erfolgt nicht. Es kann lediglich überprüft werden, ob eine gegebene Abbildung die Einhaltung der Reihenfolge erfüllt bzw. ob eine solche Abbildung überhaupt möglich ist. Das Löschen bzw. Nichtabbilden von Knoten wird ebenfalls nicht angesprochen.

Sowohl der 1:N-Ansatz von Weidlich et al. (2010a) als auch die M:N-Abbildungen von Castelo Branco et al. (2012a) und Polyvyanyy et al. (2012) erlauben keine beliebigen M:N-Abbildungen, da sie durch Nachbarschaftsbeziehungen, Knotenreihenfolgen etc. vorselektiert werden. Diese Vorselektionen sind jedoch nicht stichhaltig, sondern folgen mehr oder weniger Heuristiken, die durchaus diskutiert werden können. So ist zum Beispiel nicht klar, warum bei aufeinander abgebildeten Aktivitätenmengen die Mengenbildung nur unter Einhaltung der Reihenfolgenbedingung erfolgen darf. So könnten bei einem Modell mit der Aktivitätenfolge a: „Kaffeebohnen in Mühle füllen“ → b: „Wasser in Wasserkocher füllen“ → c: „Kaffeebohnen mahlen“ die beiden Aktivitäten a und c nicht zu einer Menge zusammengefasst werden, was unter Umständen sinnvoll sein kann, da die Reihenfolge zwischen a und b eine andere ist als zwischen c und b. Aus diesem Grund wird in Abschnitt 3.3.2 eine neue, allgemeine M:N-Abbildung definiert, die auch die Mengenbildung im eben gezeigten Beispiel zulässt.

3.3.2 Allgemeine M:N-Abbildungen

Es wird nun eine Abbildung definiert, die es erlaubt, sowohl hierarchische als auch nicht hierarchische Verfeinerungen von Prozessmodellen sowie Mischformen zu berücksichtigen, unabhängig vom Aufbau eines Prozessmodells. Diese M:N-Abbildung, die erstmals von Baumann et al. (2014) vorgeschlagen wird, ist hierbei eine natürliche Erweiterung der in Abschnitt 2.2.1.1 eingeführten 1:1-Abbildung und kann beliebige Mengen an Aktivitäten, nicht nur benachbarte Aktivitäten, aufeinander abbilden. Die Menge der Knoten eines Prozessmodells wird hierbei disjunkt und vollständig in Teilmengen zerlegt. Die Abbildung ist bezüglich dieser Teilmengen bijektiv, d. h., die Funktion ist insbesondere invertierbar. Die M:N-Abbildung wird, da die Abgleichsmethoden in Kapitel 4 stets nur eine Abbildung der Aktivitäten voraussetzen, direkt auf den Mengen der Aktivitäten definiert.

Definition 3.3 (M:N-Abbildung)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Aktivitätsmengen A_1 und A_2. Weiter sei die Menge $P_i \subseteq \mathcal{P}(N_i) \ni \emptyset$ eine vollständige, disjunkte Zerlegung von A_i, d. h.,

$$p_k \cap p_j = \emptyset \ \forall p_k, p_j \in P_i \text{ mit } p_k \neq p_j \ \wedge \bigcup_{p_k \in P_i} = A_i$$

für $i = 1, 2$. Eine M:N-Abbildung zwischen G_1 und G_2 ist gegeben durch eine bijektive Funktion M mit

$$M : P_1 \to P_2,\ p_1 \mapsto p_2 \ \forall p_1 \in P_1, p_2 \in P_2.$$

Insbesondere bedeutet $\emptyset \mapsto p_2$ bzw. $p_1 \mapsto \emptyset$, dass p_2 bzw. p_1 gelöscht, also nicht abgebildet werden; $\emptyset \mapsto \emptyset$ wird ausgeschlossen. Genauso wird ausgeschlossen, dass alle Aktivitäten gelöscht werden, also dass $P_1 = \{p_1, \emptyset\}$ und $P_2 = \{p_2, \emptyset\}$ mit $M(p_1) = \emptyset$ und $M^{-1}(p_2) = \emptyset$.

Das Ausschließen des Falls, dass alles gelöscht wird, vereinfacht viele der noch folgenden Formeln, da auf eine Fallunterscheidung verzichtet werden kann. Zudem stellt es keine Einschränkung dar, da zwei Prozessmodellen bei einer Abbildung, die alle Aktivitäten löscht, ein Ähnlichkeitswert von 0 zugewiesen wird. Jede Abbildung, die mindestens eine Aktivität auf eine andere abbildet, liefert jedoch auf keinen Fall einen schlechteren Ähnlichkeitswert und auf jeden Fall einen Ähnlichkeitswert > 0, wenn wie in Abschnitt 2.2.1.3 der Anteil der abgebildeten Knoten mit einem positiven Gewicht berücksichtigt wird.

Die Mächtigkeiten der Partitionen P_1 und P_2 der Prozessmodelle G_1 und G_1 ohne Berücksichtigung der leeren Menge sind entweder gleich oder unterscheiden sich um 1. Bei Gleichheit wird entweder aus beiden Modellen kein Knoten gelöscht (siehe Abbildung 3.2) oder aus beiden Modellen wird jeweils eine Menge an Knoten gelöscht (siehe Abbilung 3.3); bei einem Unterschied der Mächtigkeit von 1 wird nur in einem der Modelle eine Menge an Knoten gelöscht (siehe Abbildung 3.4). Es gilt mit Berücksichtigung der leeren Menge für alle Abbildungen stets: $|P_1| = |P_2|$.

Wird für abgebildete Knotenmengen, das sind die Knotenmengen, die nicht auf die leere Menge abbilden bzw. nicht das Bild der leeren Menge sind, gefordert, dass deren Mächtigkeit 1 sein muss, so handelt es sich bei einer solchen Abbildung um die 1:1-Abbildung aus Definition 2.2.

Der Abschnitt 3.4 befasst sich mit einer gemeinsamen, abstrakten Darstellung von sowohl multiperspektivischen imperativen als auch multiperspektivischen deklarativen Prozessmodellen. Diese gemeinsame Darstellung wird benötigt, um eine der in Abschnitt 3.5 genannten Methodenübertragungen zu begründen.

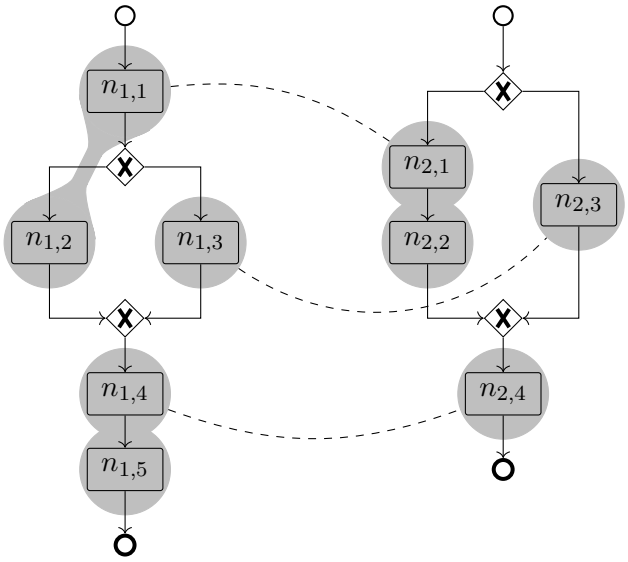


Abbildung 3.2: Beispielabbildung ohne gelöschte Aktivitäten

3.4 Gemeinsamkeiten von imperativen und deklarativen Prozessmodellen

Obwohl natürlich starke Unterschiede zwischen imperativ und deklarativ formulierten Prozessmodellen bestehen, gibt es doch auch einige Gemeinsamkeiten, die beide Modellierungsansätze auszeichnen und die bei einem Ähnlichkeitsabgleich von imperativen und deklarativen Modellen mittels Abstraktion ausgenutzt werden können. Auf der einen Seite gibt es immer eine Menge an Aktivitäten mit Aktivitätsbeschreibungen, die unabhängig von Sprache und Modellieransatz feststeht. Außerdem gibt es externe Ressourcen wie Agenten, Services und Datenobjekte, die den Aktivitäten zugewiesen sind oder während ihrer Ausführung verwendet oder aufgerufen werden können. Aktivitäten und Ressourcen sind hierbei mittels Relationen verknüpft, die sich in Inter- und Intraaktivitätsrelationen einteilen lassen. Intraaktivitätsrelationen sind Relationen, die genau eine Aktivität mit beliebig vielen Ressourcen verknüpfen, also Ressourcen den Aktivitäten zuordnen, Interaktivitätsrelationen sind Relationen, die (mindestens) zwei Aktivitäten miteinander verknüpfen, also eine Abhängigkeit von Aktivitäten angeben; diese können auch Ressourcen mit einschließen.

Auf Basis dieser sehr abstrakt gehaltenen Sichtweise auf Aktivitäten, Ressourcenmengen und Relationen, lässt sich ein generalisiertes Prozessmodell wie folgt definieren:

Definition 3.4 (Generalisiertes Prozessmodell)**.** Es sei $\mathcal{N}$ eine Menge an Aktivitäten, $\mathcal{L}$ eine Menge an Beschreibungen, $\mathcal{A}$ eine Menge an Agenten, $\mathcal{D}$ eine Menge an Datenobjekten und $\mathcal{S}$ eine Menge an Werkzeugen. Es sei zudem $\mathcal{U} = \mathcal{N} \dot\cup \mathcal{L} \dot\cup \mathcal{A} \dot\cup \mathcal{D} \dot\cup \mathcal{S}$. Ein generalisiertes Prozessmodell GPM ist von der Form

$$GPM = (\mathcal{U}, \mathcal{R})$$

wobei $\mathcal{R}$ eine Menge an Relationen ist. Eine n-äre Relation $R \in \mathcal{R}$ habe die Form $R \subseteq U_1 \times \ldots \times U_n$ mit $U_i \in \{\mathcal{N}, \mathcal{L}, \mathcal{P}(\mathcal{N} \times \mathcal{L}), \mathcal{P}(\mathcal{A} \times \mathcal{L}), \mathcal{P}(\mathcal{D} \times \mathcal{L}), \mathcal{P}(\mathcal{S} \times \mathcal{L})\}$, $i = 1, \ldots, n$.

Eine Relation könnte zum Beispiel „Sequenzfluss“ sein, eine andere beispielsweise „benötigte Daten“. Die Kombination von Beschriftungen ($\mathcal{L}$) mit Knoten, Agenten, Dokumenten und

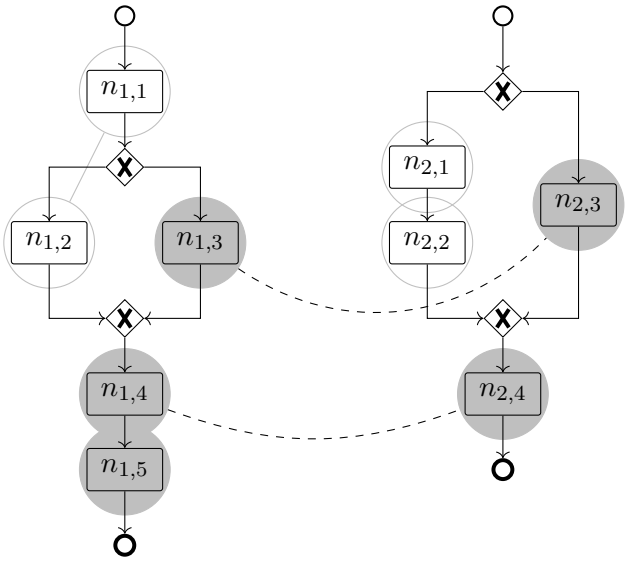


Abbildung 3.3: Beispielabbildung mit gelöschten Aktivitäten in beiden Modellen: $M(\{n_{1,1}, n_{1,2}\}) = \emptyset$, $M^{-1}(\{n_{2,1}, n_{2,2}\}) = \emptyset$

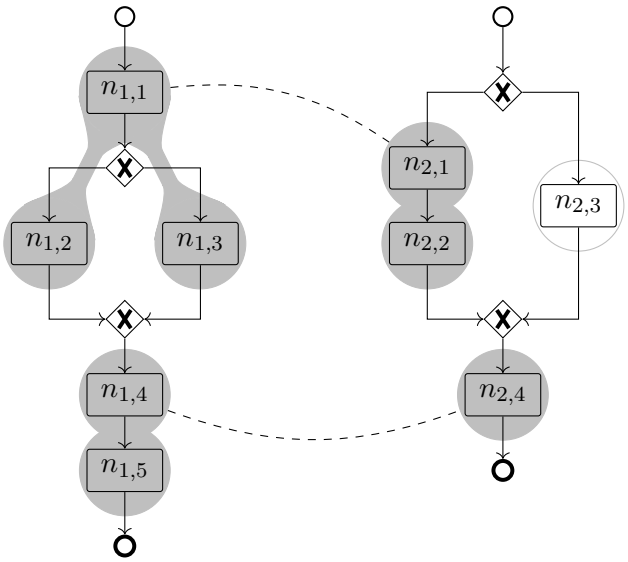


Abbildung 3.4: Beispielabbildung mit gelöschter Aktivität im rechten Modell: $M^{-1}(\{n_{2,3}\}) = \emptyset$

Services in der Definition der U_i bietet die Möglichkeit, Annotationen jeglicher Art, z. B. auch Bedingungen bei Entscheidungsknoten, zu berücksichtigen. Diese Annotationen können auch leer sein. Ereignisknoten wie das Start- und Enderegnis sind im generalisierten Prozessmodell nicht vorhanden, da sie für das Modell bzw. den Prozessablauf an sich keine zusätzlichen Informationen liefern. Für die Relationen gilt, wie oben geschrieben, folgende Unterscheidung:

Definition 3.5 (Intra- und Interaktivitätsrelationen)**.** Eine n-äre Relation $R \subseteq U_1 \times \ldots \times U_n$ ist eine Intraaktivitätsrelation (Zuordnung innerhalb einer Aktivität), falls sie genau eine Aktivität mit beliebigen Ressourcen verknüpft (Zuordnung von Ressourcen zu einer Aktivität), d. h. formal ausgedrückt

$$n \geq 2 \wedge \exists! i : U_i = \mathcal{N} \wedge \nexists j : U_j = \mathcal{P}(\mathcal{N} \times \mathcal{L}) \text{ oder } n = 1 \wedge U_1 = \mathcal{P}(\mathcal{N} \times \mathcal{L}).$$

Eine n-äre Relation R, $n \geq 2$, ist eine Interaktivitäsrelation (Zuordnung zwischen mehreren Aktivitäten), falls sie mindestens zwei Aktivitäten, möglicherweise aber auch mehr, mit beliebigen Ressourcen in Beziehung setzt. Formal gilt für diese

$$\exists i : U_i = \mathcal{P}(\mathcal{N} \times \mathcal{L}) \text{ oder } \exists j \neq k : U_j = U_k = \mathcal{N}.$$

Diese Relationen sind im Allgemeinen weder symmetrisch, noch reflexiv oder transitiv. Auch Liu et al. (2007) unterscheiden in ihrer Arbeit zwischen zwei Dimensionen beim Modellieren von Prozessen: dem Kontext und dem Verhalten. Das Verhalten spezifiziert die Reihenfolge von Aufgaben, ob Aufgaben überhaupt ausgeführt werden usw., während der Kontext spezifiziert, welche Aufgaben welche Ergebnisse liefern, welche Informationen benötigen, wie/womit sie ausgeführt werden usw. Diese Unterschiedung in Kontext und Verhalten entspricht der Unterscheidung in Intra- und Interaktivitätsrelationen, die jeweils den Kontext bzw. das Verhalten festlegen. Die folgenden Beispiele sollen die generalisierte Sichtweise auf konkrete Prozessmodelle verdeutlichen:

Beispiel 3.1. Gegeben sei das Prozessmodell aus Abbildung 3.5, das in BPMN modelliert ist und in dem vier Prozessperspektiven dargestellt sind, die funktionale, die verhaltensbasierte, die organisatorische und die datenorientierte. Die Prozessgraphdarstellung gemäß Definition 3.1 ist wie folgt: $G = (N, E, \lambda)$ mit

- $A = \{a_1, a_2, a_3, a_4\}$
- $C = \{XOR_s, XOR_j\}$
- $N = A \cup C \cup \{e_{start}, e_{end}\}$
- $E = \{(e_{start}, a_1), (a_1, XOR_s), (XOR_s, a_2), (XOR_s, a_3), (a_2, XOR_j), (a_3, XOR_j), (XOR_j, a_4), (a_4, e_{end})\}$
- $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ mit
  - $\lambda(a_1) = (\text{"A"}, \{\text{Agent 2}\}, \{\}, \{\})$
  - $\lambda(a_2) = (\text{"B"}, \{\text{Agent 2}\}, \{\}, \{\})$
  - $\lambda(a_3) = (\text{"C"}, \{\text{Agent 1}\}, \{\}, \{\})$
  - $\lambda(a_4) = (\text{"D"}, \{\text{Agent 2}\}, \{\text{Dok 1}\}, \{\})$

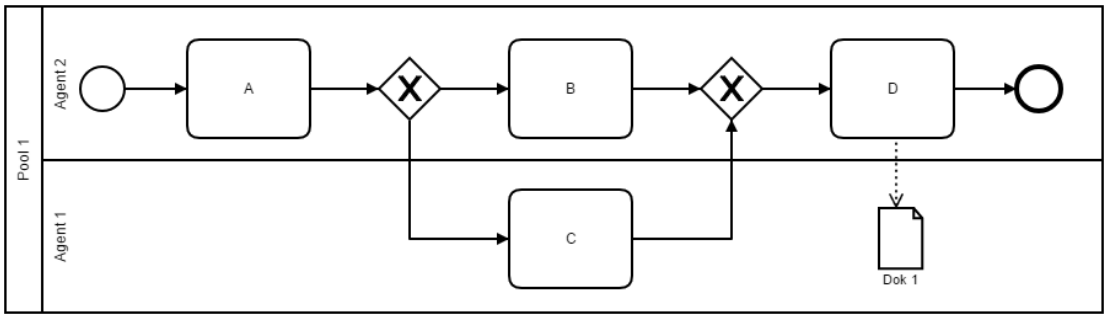


Abbildung 3.5: Ein Beispielmodell in BPMN

Die Kanten, die in E zusammengefasst sind und zwei Aktivitäten miteinander verbinden, können in der generalisierten Form als Sequenzrelationen $seq \in \mathcal{R}$ aufgefasst werden: $(n_1, n_2) \in E \triangleq sequ(n_1, n_2)$ mit $sequ \subseteq \mathcal{N} \times \mathcal{N}$ mit $A = \mathcal{N}$, wobei $sequ$ damit eine Interaktivitätsrelation ist. In obigem Beispiel gibt es zwei solche Relationsausprägungen, $sequ = \{(a_1, XOR_s), (XOR_j, a_4)\}$. Stattdessen können der XOR-Split und der XOR-Join als Relationen aufgefasst werden, die einen Knoten mit einer Menge an Knoten bzw. eine Menge an Knoten mit genau einem Knoten verknüpfen: $xors \subseteq \mathcal{N} \times \mathcal{L} \times \mathcal{P}(\mathcal{N} \times \mathcal{L})$ und $xorj \subseteq \mathcal{P}(\mathcal{N} \times \mathcal{L}) \times \mathcal{N}$. Sowohl $xors$ als auch $xorj$ sind Interaktivitätsrelationen. Für das Beispielmodell ist $xors = \{(XOR_s, "", \{(a_2, ""), (a_3, "")\})\}$ und $xorj = \{(\{(a_2, ""), (a_3, "")\}, XOR_j)\}$.

Für die Mengen an Agenten und Dokumenten gilt, dass $\mathcal{A} = \{\text{Agent 1}, \text{Agent 2}\}$ und $\mathcal{D} = \{\text{Dok 1}\}$. Die Menge der Labels ist $\mathcal{L} = \{"", "A", "B", "C", "D"\}$. Da jede Funktion auch eine Relation ist, können λ bzw. die λ_i als weitere Relationen aufgefasst werden, zum Beispiel $\lambda_1 \triangleq desc$ mit $desc \subseteq A \times \mathcal{L}$, $\lambda_2 \triangleq qual$ mit $qual \subseteq A \times \mathcal{A}$ und $\lambda_3 \triangleq requ$ mit $requ \subseteq A \times \mathcal{D}$. Dabei sind $desc$, $qual$ und $requ$ Intraaktivitätsrelationen, also Relationen, die einer Aktivität eine Menge an Ressourcen bzw. Labels zuordnen. Für das Beispielmodell ist

- $desc = \{(a_1, "A"), (a_2, "B"), (a_3, "C"), (a_4, "D")\}$,
- $qual = \{(a_1, \{\text{Agent 2}\}), (a_2, \{\text{Agent 2}\}), (a_3, \{\text{Agent 1}\}), (a_4, \{\text{Agent 2}\})\}$ und
- $requ = \{(a_4, \{\text{Dok 1}\})\}$.

Mit $\mathcal{N} = A$, $\mathcal{L}$, $\mathcal{A}$ und $\mathcal{D}$ wie angegeben und $\mathcal{S} = \emptyset$ ist das generalisierte Prozessmodell damit $GPM = (\mathcal{U}, \mathcal{R})$ mit $\mathcal{U} = \mathcal{N} \dot\cup \mathcal{L} \dot\cup \mathcal{A} \dot\cup \mathcal{D} \dot\cup \mathcal{S}$ und $\mathcal{R} = \{sequ, xors, xorj, desc, qual, requ\}$.

Beispiel 3.2. Für ein deklaratives Prozessmodell, beispielsweise für ein Modell, das in DECLARE bzw. ConDec formuliert ist, unterscheidet sich die generalisierte Darstellung kaum von der regulären. Die Mengen $\mathcal{A}$, $\mathcal{D}$ und $\mathcal{S}$ sind für DECLARE/ConDec-Modelle leer. Die einzige Intraaktivitätsrelation ist die, die einer Aktivität eine Beschreibung zuordnet. Die Interaktivitätsrelationen sind die Relationen, die in DECLARE/ConDec bereits formuliert sind und jeweils mindestens zwei Aktivitäten miteinander verknüpfen, z. B. die Relationen $response \subseteq \mathcal{N} \times \mathcal{N}$, $precedence \subseteq \mathcal{N} \times \mathcal{N}$ oder $alternateExistence \subseteq \mathcal{N} \times \mathcal{N}$ mit $\mathcal{N} = A$. Die Existenzrelationen, die angeben, wie oft eine Aktivität mindestens ausgeführt werden muss und wie oft sie maximal ausgeführt werden darf, sind nach Definition Intraaktivitätsrelationen, die jeder Aktivität eine Zahl bzw. zwei natürliche Zahlen zuordnen, z. B.

$existence \subseteq \mathcal{N} \times \mathbb{N}_0 \times \mathbb{N}_0$, wobei die Zahlen als Teil der Labelmenge $\mathcal{L}$ angesehen werden können.

Die generalisierte Sichtweise auf Prozessmodelle wird in Abschnitt 3.5 zur Übertragung von Abgleichsmethoden von imperativen Prozessmodellen auf deklarative Modelle (wobei auch die umgekehrte Richtung möglich ist) verwendet. Es gilt, dass Abgleichsmethoden, die lediglich Informationen verwenden, die in der generalisierten Sichtweise vorhanden sind, von einem Modell auf ein anderes in einer anderen Modelliersprache übertragen werden können. Dies wird in Abschnitt 3.5.3 erläutert. Die anderen genannten Übertragbarkeiten beziehen sich auf die Abbildung (Übertragbarkeit von M:N- auf 1:1-Abbildungen, Abschnitt 3.5.1) und auf die Darstellung der Ressourcen (Übertragbarkeit zwischen verschiedenen Ressourcen von Aktivitäten, Abschnitt 3.5.2). Diese Übertragbarkeiten werden zunächst allgemein festgestellt, bevor in Kapitel 4 die eigentlichen M:N-Abgleichsmethoden für die neue Definition der multiperspektivischen Prozessmodelle eingeführt und gleich mit den jeweiligen Übertragungsmöglichkeiten auf andere Bereiche/Modelle erweitert werden.

3.5 Übertragbarkeit von Abgleichsmethoden

Ähnlich wie bei der Übertragung des Editierabstandes von Zeichenketten auf Graphen (Abschnitt 2.2.3.10) können auch Berechungsmethoden für die Ähnlichkeit einer Prozessperspektive auf die Ähnlichkeitsberechnung einer anderen Perspektive übertragen werden, sofern die Struktur der Eingabevariablen für die Berechnungsmethode dieselbe ist. Beim Editierabstand sind die Eingabevariablen die Kosten für Änderungsoperationen. Ob sich diese Kosten auf die Änderung von einzelnen Zeichen in Zeichenketten oder auf Änderungen von Elementen in Graphen beziehen, spielt für die Methode an sich keine Rolle. Die Übertragbarkeit von Methoden ist insofern nützlich, als dass sie es erlaubt, bestehende Methoden auf einen, bezogen auf die Übertragbarkeit, ähnlichen Kontext anzuwenden. Die Neudefinition von Abgleichsmethoden ist dann nicht unbedingt mehr notwendig. Die unterschiedlichen Bereiche, in denen ein Abgleich erfolgen kann und zwischen denen eine Methodenübertragung vorgenommen werden kann, sind in Abschnitt 3.7 aufgeführt. Es werden im Folgenden drei verschiedene Übertragunsmöglichkeiten genannt, wobei mit der offensichtlichsten, der Übertragung aufgrund der Definition der Abbildung, begonnen wird. Es folgen die Übertragbarkeit aufgrund der Ressourcenbeschaffenheit und die Übertragbarkeit aufgrund gleicher Modellbeschaffenheit. Letztgenannte verwendet die gemeinsame Modelldarstellung aus Abschnitt 3.4.

3.5.1 Übertragbarkeit auf Basis der Abbildung

Durch die Definition der M:N-Abbildung in Definition 3.3, die, wie dort beschrieben, eine Verallgemeinerung der 1:1-Abbildung aus Definition 2.2 darstellt, lassen sich alle Methoden, die eine M:N-Abbildung zwischen Prozessmodellen voraussetzen, auch auf den Fall, dass explizit 1:1-Abbildungen gesucht werden, anwenden. Dies ist eine natürliche Folgerung aus den Definitionen. Der umgekehrte Fall ist im Allgemeinen dagegen nicht möglich.

Es ist natürlich zu erwarten, dass ein Abgleich, der explizit auf 1:1-Abbildungen beruht, bessere Ergebnisse liefert, wenn Methoden angewendet werden, die speziell auf 1:1-Abbildungen zugeschnitten sind, doch der Optimierungsansatz in Abschnitt 4.4.4 lässt standardmäßig beliebige M:N-Abbildungen zu. Ist die Art der Korrespondenzen im Voraus nicht

bekannt, so kann für einen ersten Eindruck der Ähnlichkeit zweier Prozessmodelle ein M:N-fähiger Ansatz gewählt werden. Zeichnet sich als optimale Abbildung hierbei eine 1:1-Abbildung ab, so können in einem weiteren Durchgang Abgleichsmethoden explizit für 1:1-Abbildungen angewendet werden.

3.5.2 Übertragbarkeit auf Basis der Ressourcen

Diese Übertragbarkeit betrifft die Menge der Agenten, Datenobjekte und Services, die in den multiperspektivischen Modellen der Definitionen 3.1 und 3.2 zu den bisherigen Definitionen von Prozessmodellen hinzugefügt wurden. Werden die Ressourcen (Agenten, Datenobjekte, Services) als Eigenschaften von Aktivitäten aufgefasst, siehe Abschnitt 3.2.3, und kann die für eine Abgleichsmethode auf einer der Ressourcenmengen vorausgesetzte Struktur in einer der anderen Ressourcenmengen wiedergefunden werden, so kann die Abgleichsmethode auf die andere Ressourcenmenge übertragen werden.

Setzt beispielsweise eine Abgleichsmethode für die organisatorische Perspektive eine Menge an Agenten voraus, also eine Menge an persönlichen Namen oder Rollen- bzw. Gruppennamen, ohne eine (möglicherweise vorhandene) dahinterliegende Struktur zu verwenden, und liegen Datenobjekte ebenfalls als eine Menge an (Namen der) Datenobjekte vor, so kann die Methode auf die datenorientierte Perspektive übertragen werden. Liegen die Agenten in einer hierarchischen Struktur vor und die Datenobjekte ebenfalls, und verwendet eine Abgleichsmethode auf der organisatorischen Perspektive diese Struktur, dann kann die Methode ebenfalls auf die datenorientierte Perspektive übertragen werden. Haben die Datenobjekte jedoch keine solche Struktur, ist die Methode nicht übertragbar, da sie Informationen verwendet, die in der anderen Perspektive nicht vorhanden sind.

Wie im Anschluss an Definition 3.1 erläutert, ist die Lesart von zugewiesenen Agenten und zugewiesenen Datenobjekten bzw. Werkzeugen eine andere. Bei einer Menge von Agenten sind die Elemente als exklusive Disjunktion zu verstehen, bei Mengen von Datenobjekten und Werkzeugen sind die Elemente als Konjunktion zu verstehen. Dies wirkt sich natürlich auf das Ähnlichkeitsmaß aus, jedoch nur in dem Maß, dass ein bestimmtes Ähnlichkeitsmaß für einen Anwendungsfall besser geeignet ist als ein alternatives Maß. Die Übertragbarkeit an sich wird dadurch nicht beeinflusst, wie in Abschnitt 4.2 am konkreten Beispiel erkennbar ist.

3.5.3 Übertragbarkeit auf Basis von Modellgemeinsamkeiten

Die Übertragbarkeit auf Basis von Modellgemeinsamkeiten baut auf der Definition der generalisierten Prozessmodelle (Definition 3.4) auf. Abgleichsmethoden, die lediglich diese abstrakte Definition von Prozessmodellen benötigen, können so auf beliebige Modelle übertragen werden. Insbesondere ist es hier auch möglich, Methoden von imperativen auf deklarative Prozessmodelle oder umgekehrt zu übertragen. Abgleichsmethoden, für die diese Übertragbarkeit möglich ist, bauen auf den Aktivitäten und ihren Beschreibungen sowie auf den Relationen auf. Aktivitäten werden über eine Funktion aufeinander abgebildet, wobei die Ähnlichkeit von Aktivitätenbeschreibungen mit bekannten labelbasierten Methoden untersucht werden kann, und anschließend können die Relationen miteinander verglichen werden. Die Relation *seq* aus Beispiel 3.1, die einen Sequenzfluss im BPMN-Modell beschreibt, könnte beispielsweise eine große Ähnlichkeit, wenn auch keine Gleichheit, zur Relation *response* aus Beispiel 3.2 haben, bei der die Ausführung einer Aktivität die spätere Ausführung einer anderen bedingt. Wichtiger allerdings als bei Interaktivitätsrelationen ist die Übertragbarkeit auf Basis von Modellgemeinsamkeiten für die Intraaktivitätsrelationen, also diejenigen Relationen, die den

Aktivitäten Agenten, Datenobjekte und Werkzeuge zuordnen. Für einige der in Abschnitt 4.2 vorgestellten Abgleichsmethoden gilt diese Übertragbarkeit.

3.6 Identifikation der konkreten Anwendungsfelder von Abgleichsmethoden

Anhand der bisherigen Vorarbeit lassen sich drei grundsätzlich orthogonale Unterscheidungskriterien für Abgleichsmethoden von Prozessmodellen finden. Abgleichsmethoden müssen an die verwendeten Prozessperspektiven (funktional, datenbasiert, operational, organisatorisch, verhaltensbasiert; zusätzlich auch an die Struktur), an die Art der zugrunde gelegten Abbildung der Modellelemente sowie an die Art des Prozessmodells an sich angepasst sein. Diese drei orthogonalen Unterscheidungskriterien sind in Abbildung 3.6 angedeutet. Jede Abgleichsmethode deckt (mindestens) eine der in Abschnitt 3.2.1 vorgestellten Perspektiven ab, wobei hier noch die zusätzliche Klasse der strukturbasierten Abgleichsmethoden gegeben ist. Korrespondenzen zwischen Aktivitäten, die gefunden werden sollen, können 1:1, 1:N oder M:N sein. Außerdem kann eine Abgleichsmethode für den Vergleich zweier imperativer Prozessmodelle (imp-imp), zweier deklarativer Prozessmodelle (dekl-dekl) oder zweier beliebiger Modelle, insbesondere also auch eines imperativen und eines deklarativen (imp-dekl), verwendet werden.

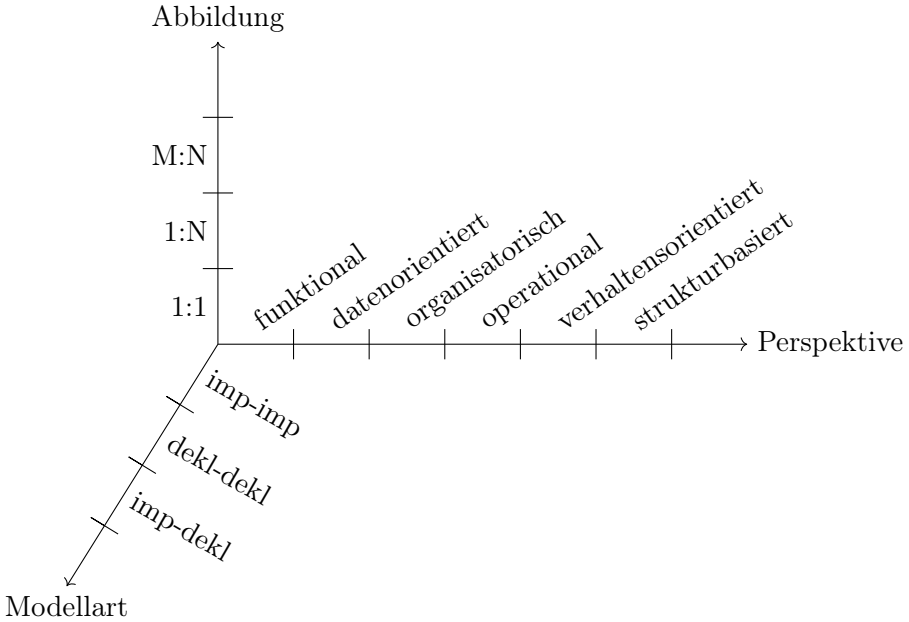


Abbildung 3.6: Orthogonale Unterschiedungskriterien für Abgleichsmethoden

Diese Aufteilung von Abgleichsmethoden in die drei orthogonalen Kategorien *Perspektive, Abbildung* und *Modellart* erlaubt auch ein strukturiertes Vorgehen bei der Auswahl von Abgleichsmethoden. Ein mögliches Vorgehen kann anhand von Abbildung 3.6 in drei Schritten erfolgen, die nicht notwendigerweise in der angegebenen Reihenfolge durchgeführt werden müssen.

Bestimmen der Modellart Die Modellart ist durch die Modelle, die abgeglichen werden sollen, von außen vorgegeben. Werden zwei imperative Modelle miteinander verglichen,

muss eine Methode gewählt werden, die imperative Modelle unterstützt. Für einen Abgleich von zwei deklarativen Prozessmodellen, muss eine Methode gewählt werden, die deklarative Modelle unterstützt. Soll ein imperatives mit einem deklarativen Prozessmodell verglichen werden, muss eine Methode gewählt werden, die beide Arten gleichzeitig unterstützt. Dies ist in der Regel der Fall, wenn die Methode auf der generalisierten Modelldarstellung aus Definition 3.4 arbeitet. Die Modellart sollte hierbei für jedes Modellpaar überprüft werden, wobei die Modelle in einem Repositorium in der Regel vom selben Typ und sogar in derselben Sprache modelliert sein sollten. Dies muss jedoch nicht der Fall sein. Für imperative Prozessmodelle gilt, dass sie in der Form von Definition 3.1 vorliegen, und deklarative Modelle liegen in der Form von Definition 3.2 vor. Das heiß, für die Abgleichsmethoden existiert jeweils genau eine Art imperative bzw. deklarative Modelldefinition. Befinden sich in einem Repositorium mehrere verschiedene Arten von Modellen, beispielsweise BPMN-Modelle und EPKs, so werden diese bereits beim Einlesen in Prozessgraphen transformiert.

Bestimmen der Prozessperspektiven Ebenfalls von außen vorgegeben sind die unterschiedlichen Prozessperspektiven. So gut wie jede Prozessmodelliersprache modelliert die funktionale Perspektive, also die Aktivitäten. Auch die verhaltensorientierte Perspektive bzw. die Struktur ist üblicherweise in den Modellen abgebildet. Die weiteren Perspektiven hängen stark von der zugrunde liegenden Modelliersprache ab, unabhängig davon, ob es sich dabei um imperative oder deklarative Sprachen handelt. BPMN bietet beispielsweise keine Möglichkeit, die operationale Perspektive darzustellen. Abgesehen von den Perspektiven in den einzelnen Modellen muss auch darauf geachtet werden, welche Perspektiven von allen zu vergleichenden Prozessmodellen abgebildet werden. Wird eine Menge an Prozessmodellen verglichen, ist es sinnvoll, nur die Perspektiven bei dem Abgleich zu berücksichtigen, die in allen Modellen vorhanden sind. Andernfalls sind die Ähnlichkeitswerte nicht vergleichbar. Es kann also passieren, dass obwohl ein Modell eine bestimmte Perspektive enthält, diese bei einem Abgleich nicht berücksichtigt wird, weil sie nicht in allen anderen Modellen enthalten ist. Eine weitere Reduktion der berücksichtigten Perspektiven kann dann erfolgen, wenn bestimmte Perspektiven zwar prinzipiell vorhanden sind, aber nicht die erforderliche bzw. angestrebte Qualität aufweisen. Diese Entscheidung ist jedoch nicht wie die bisherigen von außen vorgegeben, sondern eine eher subjektive. So kann zum Beispiel bekannt sein, dass zwar einige Datenobjekte in den Modellen richtig modelliert sind, die Perspektive jedoch nicht vollständig ist, da einige andere Datenobjekte in den Modellen fehlen. Ob sie dabei fehlen, weil sie zum Beispiel nicht in elektronischer Form vorliegen, oder weil bei der Modellierung diese Perspektive vernachlässigt wurde, spielt dabei keine Rolle.

Bestimmen der Abbildungsart Eine Entscheidung bezüglich der angestrebten Abbildung kann im Vorab getroffen, muss es jedoch nicht. Da per Definition die M:N-Abbildung aus Definition 3.3 eine Generalisierung der 1:1-Abbildung darstellt, findet also bei der Wahl der M:N-Abbildung keinerlei Einschränkung bezüglich der möglichen, zustandekommenden Korrespondenzen statt. Allerdings bedingt die Wahl der Abbildung die Menge der infrage kommenden Abgleichsmethoden. Methoden, die M:N-Korrespondenzen unterstützen, sind in der Regel nicht so spezialisiert wie Methoden, die nur einzelne Aktivitätenpaare erlauben. Es kann also eine Inspektion der abzugleichenden Modelle im Voraus erfolgen, um festzustellen, ob eine Einschränkung auf 1:1-Abbildungen gerechtfertigt ist. Diese Entscheidung ist meist jedoch rein subjektiv. Ein anderer Weg besteht darin, zunächst M:N-Abbildungen zuzulassen und die resultierenden Korrespondenzen zu untersuchen. Werden hier nur 1:1-Paare

gefunden, kann ein erneuter Abgleich durchgeführt werden, wobei dann von vornherein nur 1:1-Abbildungen zugelassen werden, um die Ähnlichkeitswerte aus der ersten Runde möglicherweise zu verbessern und Feinheiten aufzudecken. Ob hierbei tatsächlich eine Verbesserung eintritt, kann nicht garantiert werden. Falls im Vorab allerdings bekannt ist, dass auf jeden Fall 1:1-Abbildungen vorliegen, zum Beispiel weil untersucht werden soll, ob die Übersetzung eines imperativen Modells in ein deklaratives korrekt ist, kann von vornherein auf 1:1-Abbildungen zurückgegriffen werden. In diesem Fall wäre die Abbildung durch die Aktivitäten an sich auch schon festgelegt und es müsste nur noch ein Ähnlichkeitswert für die vorgegebene Abbildung errechnet werden.

Als nächstes soll überprüft werden, inwieweit der dreidimensionale Würfel mit den Kanten *Perspektive, Abbildung* und *Modellart* mit den Abgleichsmethoden aus der Literatur (Kapitel 2) abgedeckt wird. Abbildung 3.7 zeigt die Abdeckung mit den Methoden aus der Literatur, wobei der Würfel als geschachtelte Tabelle dargestellt ist. Die Abdeckung ist hierbei nicht rein quantitativ vorgenommen, sondern stellt einen qualitativen Indikator dar, der die Anzahl der unterschiedlichen Abgleichsansätze, deren Verbreitung und Güte, vor allem was die Einschränkungen ihrer Einsetzbarkeit betrifft, berücksichtigt. Diese Zahlen in Klammern stellen, nach bestem Wissen der Autorin, einen groben Richtwert für die Anzahl relevanter Methoden dar, ohne dabei die Methoden auf Basis von M:N-Abbildungen für multiperspektivische Modelle zu berücksichtigen, die in Kapitel 4 noch vorgestellt werden. Die Abdeckungsgrade reichen hierbei von


nicht vorhanden (0),


sehr gering vorhanden $(1-2)$,


gering vorhanden $(3-5)$,


vorhanden $(6-10)$ bis


ausreichend vorhanden (>10).

Auf 1:N-Abbildungen soll in Kapitel 4 und der restlichen Arbeit nicht separat eingegangen werden. Sie sind, wie auch 1:1-Abbildungen, in Definition 3.3 eingeschlossen, jedoch bieten sich explizite 1:1-Abbildungen dann an, wenn ein Abgleich mit dem Ziel besserer Modellverständlichkeit (siehe Abschnitt 1.1.3) oder zu Evaluationszwecken (siehe Abschnitt 1.1.4) durchgeführt werden soll, es also für jede Aktivität im einen Modell eine korrespondierende im anderen Modell gibt und umgekehrt. Aus diesem Grund sind 1:1-Abbildungen in Abbildung 3.7 für alle Abgleichsmöglichkeiten aufgeführt. Da es für Abgleiche imperativer Prozessmodelle Methoden gibt, die für 1:N-Abbildungen geeignet sind, nicht jedoch für M:N-Abbildungen, ist dort die Spalte 1:N zusätzlich notiert.

Die Methoden, die in der Spalte deklarativ-deklarativ berücksichtigt sind, sind in Abschnitt 4.5.1 genauer erläutert. Für den Abgleich von imperativen mit deklarativen Modellen gibt es bisher in den Arbeiten anderer Autoren keine Methodenvorschläge, genauso wenig wie für die operationale Perspektive, die vielfach bei der Definition von Prozessmodellen ausgeschlossen wird (La Rosa et al., 2011).

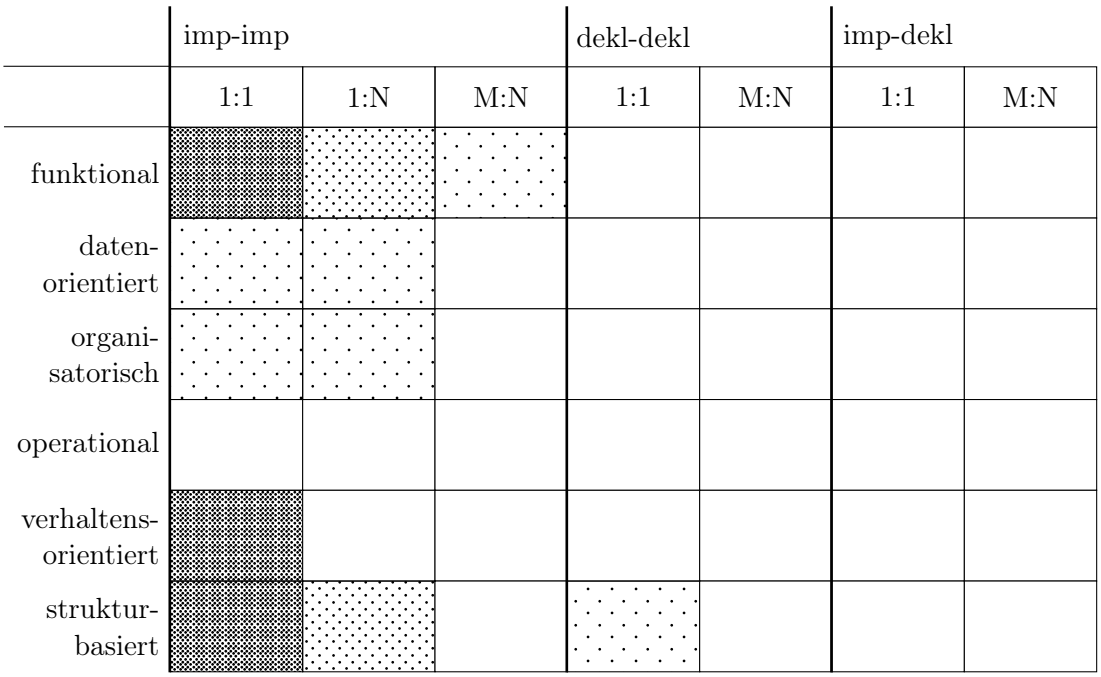


Abbildung 3.7: Anwendungsfelder und bisherige Methodenvielfalt pro Feld.

3.7 Konkrete Methodenübertragbarkeit zwischen Anwendungsfeldern

Die drei unterschiedlichen Übertragbarkeitsmechanismen von Abgleichsmethoden, die in Abschnitt 3.5 vorgestellt sind, lassen sich, wie in den Abbildungen 3.8, 3.9 und 3.10 dargestellt, auf die Anwendungsfelder übertragen. In Abbildung 3.8 ist die Übertragbarkeit auf Basis von Abbildungen dargestellt (siehe Abschnitt 3.5.1). Hierbei ist eine Methodenübertragung immer in Pfeilrichtung von dem jeweils angegebenen Anwendungsfeld in das Zielfeld gegeben. Alle Methoden, die für M:N-Abbildungen (1:N-Abbildungen) geeignet sind, funktionieren auf für 1:1- und 1:N-Abbildungen, da diese spezielle M:N-Abbildungen sind. Umgekehrt ist eine Übertragung im Allgemeinen nicht möglich.

Wie in Abschnitt 3.5.2 erwähnt, ist es möglich, je nachdem in welcher Struktur die Ressourcen vorliegen und inwieweit eine Abgleichsmethode diese Strukturen nutzt, Methoden von einer Ressourcenperspektive auf eine andere zu übertragen. Dies ist in Abbildung 3.9 dargestellt. Im Gegensatz zur Übertragbarkeit auf Basis der Abbildung, die per Definition einer Abbildung gegeben ist, ist die Übertragbarkeit auf Basis der Ressourcen nicht unbedingt für jede Abgleichsmethode möglich. Die Strukturen, in der die jeweiligen Ressourcen vorliegen, müssen, wenn sie vom Ähnlichkeitsmaß zur Berechnung verwendet werden, dieselben sein. Ist dies nicht der Fall, so kann die Methode auch nicht übertragen werden.

Setzt eine Abgleichsmethode nur solch allgemeine Strukturen voraus, die über die Definition des generalisierten Prozessmodells abbildbar sind, dann können Methoden auf unterschiedliche Modelltypen angewendet werden. Methoden, die für den Abgleich imperativer Prozessmodelle entwickelt werden, können so auch für den Abgleich deklarativer Modelle verwendet werden oder auch für den Abgleich imperativer Modelle mit deklarativen. Das heißt, dass beispielsweise eine Abgleichsmethode für die funktionale Perspektive, die eine

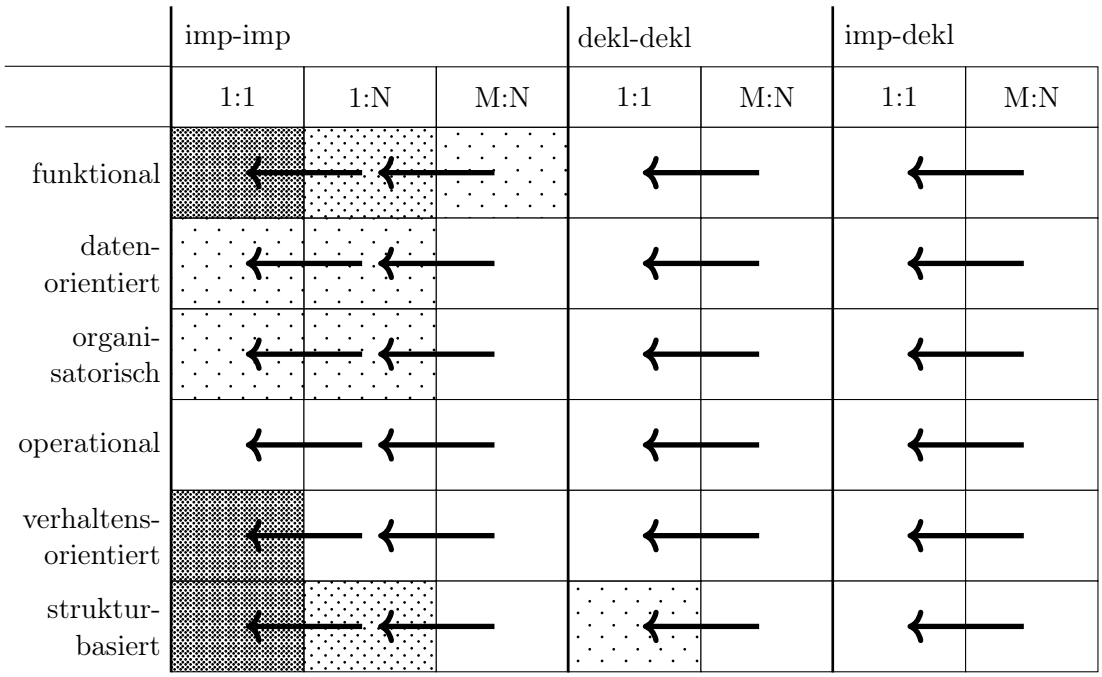


Abbildung 3.8: Anwendungsfelder und Methodenübertragbarkeit auf Basis der Abbildung.

M:N-Abbildung voraussetzt, sowohl für den Fall imp-imp, als auch für die Fälle dekl-dekl und imp-dekl herangezogen werden kann, vorausgesetzt, die nutzt nur die Information, die in der generalisierten Modelldefinition der jeweiligen Modelle zur Verfügung gestellt wird. Dies ist in Abbildung 3.10 angedeutet, wobei aus Gründen der Übersichtlichkeit die Übertragungspfeile nur für die funktionale Perspektive aufgezeichnet sind. Da sich imperative und deklarative Modelle vor allem in der Beschreibung des Verhaltens voneinander unterschieden, wird für diese Perspektive eine eher geringe Übertragbarkeit bezogen auf die Modellgemeinsamkeiten zu erwarten sein. Wiederum eine bessere Übertragbarkeit lässt die datenorientierte, die organisatorische bzw. die operationale Perspektive erwarten. Die strukturbasierten Methoden stellen eine Besonderheit dar, denn sie bauen zumeist auf der grafischen Darstellung der Prozessmodelle auf. Da deklarative Prozessmodelle jedoch häufig textuell sind, auch wenn es beispielsweise für DECLARE die grafische Notation ConDec gibt, ist für deklarative Prozessmodelle ein strukturbasierter Abgleich kaum möglich. Deswegen werden Abgleichsmethoden, die die Struktur eines Prozessmodells verwenden, im Weiteren nicht anvisiert. Überdies ist die Struktur keine der Prozessperspektiven, da sie lediglich ein Resultat der Darstellung eines Prozesses ist und nicht zum Prozess an sich gehört.

Aus diesen Überlegungen der Übertragbarkeit von Methoden ergibt sich, dass wenn für alle orthogonalen Unterscheidungskriterien (siehe Abbildung 3.6) Abgleichsmethoden zur Verfügung stehen sollen, nicht für jede Kriterienkombination eine eigene Methode gefunden werden muss. Vielmehr sollen Methoden nur für einige Kombinationen definiert werden, sodass diese für andere Kriterienkombinationen wiederverwendet werden können. Konkrete Felder, für die Methoden definiert werden können, sodass damit, d. h. nach Übertragung der Methoden, Abgleiche über den gesamten Würfel möglich sind, sind in Abbildung 3.11 mit Ausrufezeichen eingezeichnet. Die Felder in der verhaltensorientierten Perspektive sind deshalb für alle drei Modelltypen (imp-imp, dekl-dekl und imp-dekl) in der M:N-Spalte markiert, da hier frag-

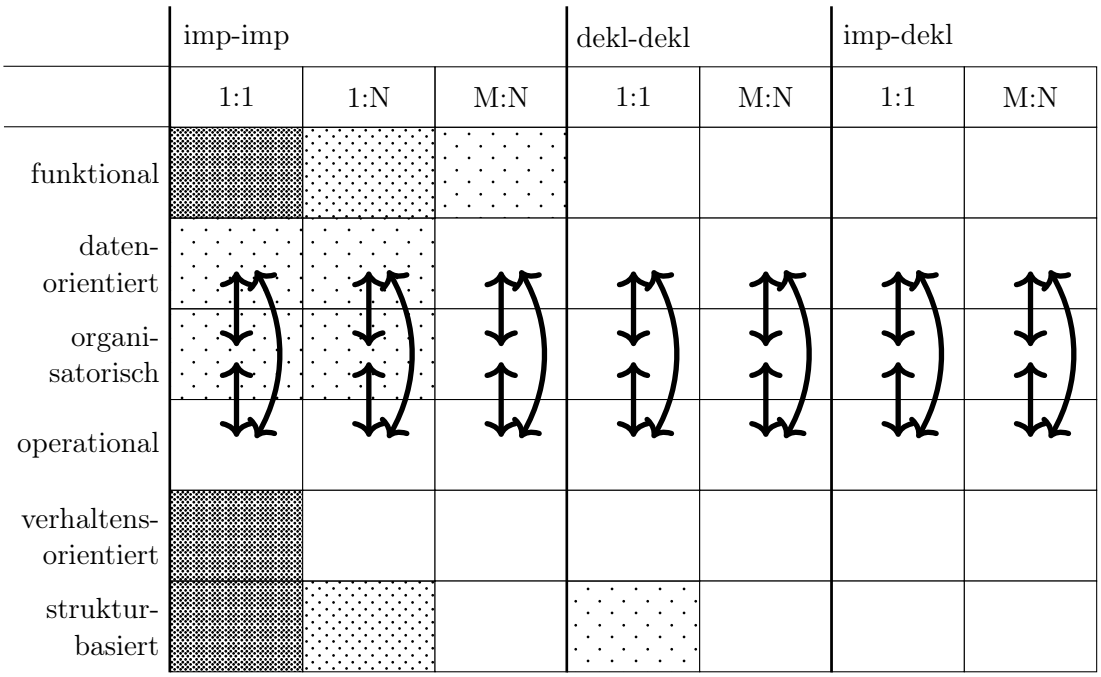


Abbildung 3.9: Anwendungsfelder und Methodenübertragbarkeit auf Basis der Ressourcen.

lich ist, inwieweit eine Übertragung der Methoden auf andere Modelltypen durch Ausnutzen der generalisierten Sichtweise möglich ist. Anstatt der organisatorischen Perspektive hätte auch eine der beiden anderen Ressourcenperspektiven in der Spalte (imp-imp, M:N) gewählt werden können, jedoch ist das organisatorische Modell in verwandten Arbeiten gut erforscht und strukturiert, sodass hier ein Ansatzpunkt relativ einfach möglich ist. Außerdem wurde die M:N-Spalte des imp-imp-Vergleichs als Ausgangspunkt ausgewählt, da zu imperativen Modellen bisher die meisten Abgleichsmöglichkeiten in der Literatur existieren und auch, da das Verständnis imperativer Modelle bzw. der Umgang mit solchen Modellen besser bzw. einfacher ist.

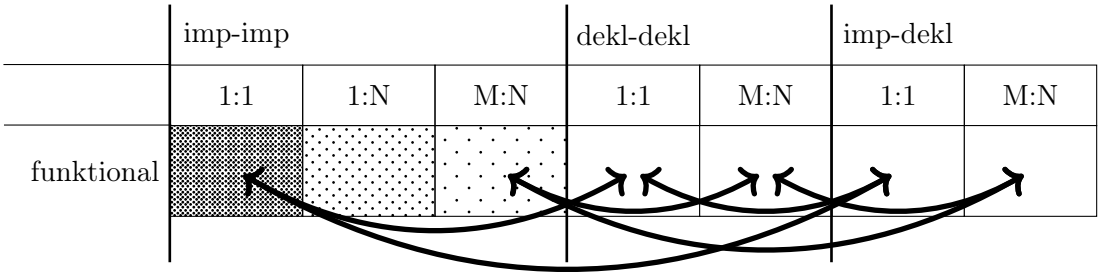


Abbildung 3.10: Anwendungsfelder und Methodenübertragbarkeit auf Basis der Modellgemeinsamkeiten.

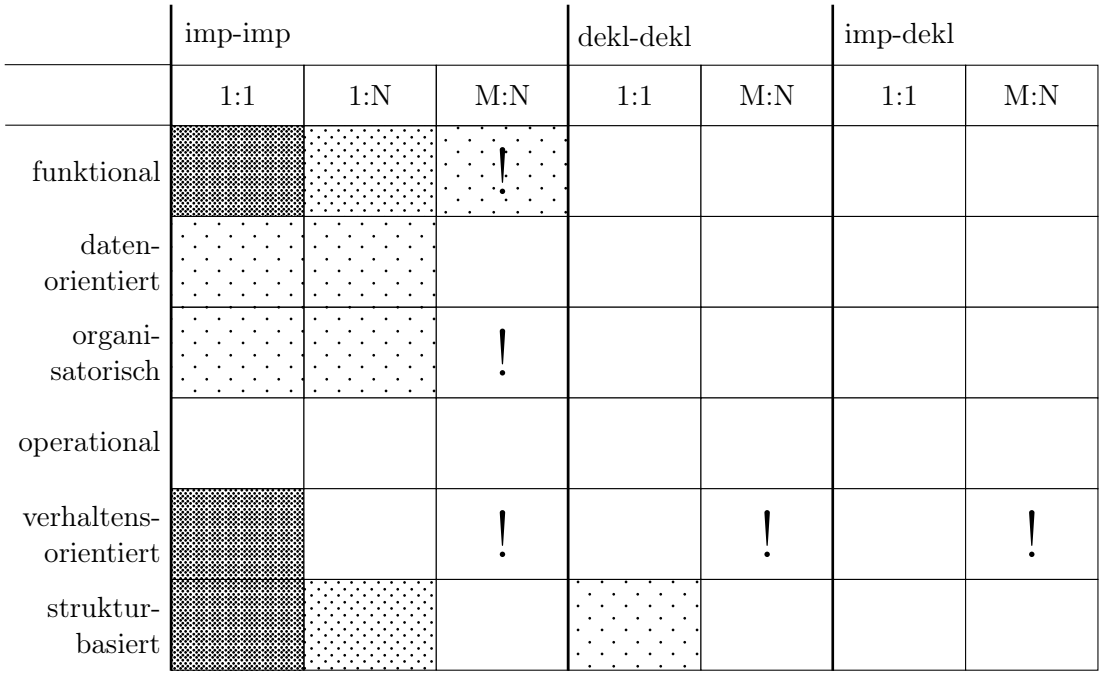


Abbildung 3.11: Ansatzpunkte bzw. Kriterienkombinationen für neue Abgleichsmethoden.

Kapitel 4

Methoden für einen M:N-Ähnlichkeitsabgleich auf multiperspektivischen Prozessmodellen

In diesem Kapitel werden Abgleichsmethoden für die in Abschnitt 3.7 identifizierten Anwendungsfelder entwickelt. Diese Methoden erlauben, wie in Abbildung 3.11 ersichtlich, M:N-Abbildungen zwischen den zu vergleichenden Prozessmodellen. Die Methoden werden hierbei separat für jede Prozessperspektive entwickelt, was aufgrund der Orthogonalität der Prozessperspektiven (Jablonski und Bussler, 1996) möglich ist. Gleichzeitig wird auch die Übertragbarkeit von Methoden gemäß Abschnitt 3.5 ausgenutzt. Bestehende Abgleichsmethoden für 1:1-Abbildungen werden, wenn möglich, adaptiert. Grundlagen der im Folgenden vorgestellten Methoden liefern die Arbeiten der Autorin, die in Zusammenarbeit mit den entsprechenden Co-Autoren entstanden sind (Baumann et al., 2014, 2015a,b, 2016a,c; Baumann, 2017). Darüber hinaus werden auch neue, bisher unveröffentlichte Methoden präsentiert. Es ist bei der jeweiligen Methode angegeben, für welche Art von Prozessmodell sie anwendbar ist, also ob insbesondere die Übertragbarkeit aufgrund der Modellart (siehe Abschnitt 3.5.3) möglich ist. Die entwickelten Ähnlichkeitswerte für die verschiedenen Perspektiven werden, analog zum Vorgehen aus der Literatur, wie im vierstufigen Ansatz aus Abschnitt 2.2.1 zu einem Ähnlichkeitswert kombiniert, der durch Bestimmen der besten Abbildung maximiert wird. Dies wird im Anschluss an die Ähnlichkeitsberechnungen der einzelnen Perspektiven gezeigt. Anschließend wird eine Überführung der Methoden auf deklarative Prozessmodelle durchgeführt.

Abschnitt 4.1 stellt ein Ähnlichkeitsmaß der Aufgabenbeschreibungen für eine M:N-Abbildung der Modelle vor. Da für dieses Anwendungsfeld in der Literatur schon ein paar wenige, andere Methoden existieren (Abbildung 3.7), ist dieser Abschnitt relativ kurz gehalten. Neben einem Ähnlichkeitsmaß, das genau die Art M:N-Abbildung aus Definition 3.3 unterstützt, wird eine kurze Diskussion von M:N-Abgleichsmethoden, die sich bereits in der entsprechenden Literatur finden, durchgeführt. Da es sich bei diesen um M:N-basierte Methoden handelt, sind sie in Abschnitt 2.2.2 nicht aufgeführt. Abschnitt 4.2 beschäftigt sich anschließend mit den drei Ressourcenperspektiven – der organisatorischen, der operationalen und der datenorientierten Perspektive. Für keine dieser drei Perspektiven existiert bislang eine Methode für einen M:N-Ähnlichkeitsabgleich. Für die operationale Perspektive lässt sich sogar überhaupt keine

Methode in der Literatur finden. Die 1:1- bzw. 1:N-Methoden für die organisatorische und datenorientierte Perspektive, die auch in Abbildung 3.7 eingezeichnet sind, werden in diesem Zusammenhang ebenfalls kurz angesprochen. Für die in Abschnitt 4.2 entwickelten Methoden findet insbesondere die Ressourcenübertragbarkeit (Abschnitt 3.5.2) Anwendung, weswegen diese drei Perspektiven auch im selben Abschnitt behandelt werden. In Abschnitt 4.3 werden Ähnlichkeitsmaße für einen M:N-Abgleich auf Basis des Modellverhaltens besprochen. Das Verhalten eines Prozessmodells wird hierbei in verschiedene Teilaspekte zerlegt, in Position, Wiederholbarkeit und Optionalität von Aktivtäten, für die getrennt voneinander ein Ähnlichkeitswert berechnet werden kann. Zusätzlich zu dieser Betrachtung, die das Verhalten auf Eigenschaften von Aktivitäten reduziert, wird ein Ähnlichkeitsmaß auf Basis von Ordnungsrelationen definiert und eines, das Flussabhängigkeiten verwendet. Letztgenanntes setzt lediglich 1:1-Abbildungen voraus, kann jedoch auf deklarative Prozessmodelle unter Anwendung der generalisierten Sichtweise auf Prozessmodelle (Abschnitt 3.5.3) übertragen werden. Abschnitt 4.4 verallgemeinert anschließend den vierstufigen Ansatz aus der Literatur (Abschnitt 2.2.1) für einen M:N-Ähnlichkeitsabgleich multiperspektivischer Prozessmodelle. Hierfür müssen vor allem der Anteil der abgebildeten Kanten und die zu maximierende Zielfunktion angepasst werden. Im Anschluss wird der Abgleich deklarativer Prozessmodelle in Abschnitt 4.5 vorgestellt, wobei versucht wird, die Methoden, die für imperative Modelle geeignet sind, zu übertragen, um vor allem auch imperative und deklarative Modelle miteinander vergleichen zu können. Das Kapitel schließt in Abschnitt 4.6 mit einer Einordnung der neu entwickelten Methoden in die Anwendungsfeldmatrix aus Abschnitt 3.7 und zeigt, wie diese mit Hilfe der verschiedenen Übertragbarkeiten gefüllt werden kann.

4.1 Labelbasierte Abgleichsmethoden

Die labelbasierten Abgleichsmethoden sind im Falle von 1:1-Abbildungen bereits sehr gut untersucht und somit wäre eine Rückführung der für M:N-Abbildungen benötigten Abgleiche auf bestehende Ansätze wünschenswert.

Eine einfache Möglichkeit, die Beschriftungen von Mengen von Aktivitäten auf einzelne Beschriftungen zurückzuführen, ist die Konkatenation dieser Beschriftungen zu einer einzelnen Zeichenkette (Baumann et al., 2014). Da die Länge der Zeichenketten im 1:1-Fall nicht beschränkt ist und die in Abschnitt 2.2.2 vorgestellten Methoden ebenfalls keine Beschränkungen voraussetzen, ist die Konkatenation eine gültige und zugleich einfache Möglichkeit, Mengen von Aktivitäten hinsichtlich ihrer Aufgabenbeschreibung zu vergleichen. Das Ähnlichkeitsmaß von Mengen von Aktivitäten ist dann wie folgt definiert.

Definition 4.1 (Erweiterte Labelähnlichkeit für Knotenmengen)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle und M eine M:N-Abbildung zwischen den beiden Modellen, $M : G_1 \to G_2$, die die Aktivitätspartitionen P_1 von G_1 und P_2 von G_2 induziert. Weiter seien $p_1 \in P_1$ und $p_2 \in P_2$ zwei Mengen an Aktivitäten mit $p_2 = M(p_1)$ und $p_1 = \{n_1, \ldots, n_k\}$ sowie $p_2 = \{m_1, \ldots, m_l\}$. Es seien $s_i = \lambda_{1,1}(n_i)$, $i = 1, \ldots, k$ und $t_j = \lambda_{2,1}(m_j)$, $j = 1, \ldots, l$ die Aktivitätenbeschreibungen der jeweiligen Aktivitäten. Die Konkatenation der Beschreibungen, ausgedrückt durch einen Punkt (.), mit einem Leerzeichen zwischen den verschiedenen Beschreibungen, ausgedrück durch $\wedge$, liefert die zusammengesetzten Aktivitätsbeschreibungen

$$\tilde{s} = s_1 . \wedge \ldots \wedge . s_k$$

und

$$\tilde{t} = t_1 ._\wedge \ldots ._\wedge .t_l.$$

Die Ähnlichkeit von p_1 und p_2 bezüglich der funktionalen Perspektive, Sim_{funk} ist dann

$$Sim_{funk}(p_1, p_2) := sim_*(\tilde{s}, \tilde{t}),$$

wobei sim_* für ein Ähnlichkeitsmaß aus Abschnitt 2.2.2 steht, das als Eingabeparameter Strings benötigt, beispielsweise sim_{sed}. Auch die Bag of Words-Ähnlichkeit sim_{bow} und die Bag of Words-Ähnlichkeit mit Label Pruning sim_{lp} aus den Abschnitten 2.2.2.5 bzw. 2.2.2.6 können auf Mengen von Aktivitäten erweitert werden, also auf $sim_{bow}(p_1, p_2)$ beziehungsweise auf $sim_{lp}(p_2, p_2)$, wenn die Funktion z, die die Beschriftungen auf die Menge ihrer Wörter im ursprünglichen String abbildet, die konkatenierten Beschriftungen von p_1 und p_2 als Eingabewerte bekommt.

Die wichtigste Frage, die sich bei dieser Lösung stellt, ist die, in welcher Reihenfolge die Beschriftungen der zu einer Menge zusammengefassten Aktivitäten konkateniert werden sollen. Die Regel, die Strings in der Reihenfolge der Aktivitäten im Prozessmodell hintereinander zu hängen, funktioniert dann nicht, sobald nicht nur sequentiell angeordnete Aktivitäten in einer Menge zusammengefasst werden. Wegen dieses Problems ist es notwendig, Stemming (siehe Abschnitt 2.2.2.2) auf die konkatenierte Aufgabenbeschreibung anzuwenden, bevor ein Ähnlichkeitsmaß darauf angewendet wird. Werden beim Stemming die verbliebenen Worte alphabetisch sortiert, kann die Reihenfolge bei der Konkatenation beliebig gewählt werden. Außerdem können mehrfach vorkommende Worte im konkatenierten String auf eine Nennung reduziert werden. Eine beliebige Konkatenationsreihenfolge gilt auch, wenn als Ähnlichkeitsmaß eine Bag of Words-Ähnlichkeit verwendet wird. Es werden hierbei Mengen an Wörtern betrachtet, für die per Definition die Reihenfolge der Wörter keine Rolle spielt. Gerade die Version mit Label Pruning ist, wie von Klinkmüller et al. (2013) beschrieben, für Labels unterschiedlicher Wörterzahl gut geeignet, wie in Abschnitt 2.2.2.5 bzw. 2.2.2.6 geschrieben.

In der Literatur lassen sich einige wenige Ansätze finden, wie Aktivitätenbeschreibungen unter M:N- oder zumindest unter 1:N-Abbildungen abgeglichen werden können. Diese Methoden sind in Abschnitt 2.2.2 nicht aufgeführt, da sie explizit über 1:1-Abbildungen hinausgehen. Allerdings sind für die angesprochenen Methoden keine ausführlichen Definitionen gegeben sondern vielmehr textuelle Beschreibungen, sodass die genaue Vorgehensweise und die Voraussetzungen für eine Anwendung nicht ganz klar sind.

So nennen Weidlich et al. (2010a) eine Möglichkeit, Aktivitätenbeschreibungen unter 1:N-Abbildungen miteinander zu vergleichen. Die Beschreibungen der einzelnen Aktivitäten werden zunächst in die Menge ihrer Wörter transformiert. Diese Mengen werden dann anschließend vereinigt und auf diesen vereinigten Mengen Abgleiche durchgeführt. Diese Methode lässt sich auch unter M:N-Abbildungen anwenden. Syntaktische Stringabgleichsmethoden wie z. B. die Levenshtein-Ähnlichkeit sim_{sed} aus Abschnitt 2.2.2.1 lassen sich dann allerdings nicht anwenden.

Die Methode, die von Castelo Branco et al. (2012a) vorgeschlagen wird, führt ebenfalls die Namen (Beschreibungen) verschiedener Aktivitäten, die über die Abbildung zusammengefasst werden, mittels Konkatenation zusammen. Allerdings werden hier zusätzlich noch die Typen der zusammengefassten Knoten, die über numerische Kürzel identifiziert sind, in die Konkatenation aufgenommen. Zur Entscheidung, ob eine Menge von Knoten ähnlich zu einer anderen Menge von Knoten ist, muss außerdem ein gewisser Anteil exakt gleicher Knoten (bezogen auf die Beschreibung und den Typ) in beiden Mengen vorhanden sein. Da in der

M:N-Abbildung, die in dieser Arbeit definiert wird, nur Aktivitäten aufeinander abgebildet werden, erübrigt sich beim Vergleich der Beschreibungen der Typvergleich. Diese 1:N- bzw. M:N-Abgleichsmethoden der funktionalen Perspektive sind in Abbildung 3.7 in den entsprechenden Zellen erfasst.

Für zwei Prozessmodelle G_1 und G_2 ist die Labelähnlichkeit unter Abbildung M dann die gemittelte Labelähnlichkeit der abgebildeten Knotenpaare:

Definition 4.2 (Labelähnlichkeit)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(A_1)$ und $P_2 \subseteq \mathcal{P}(A_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Die Labelähnlichkeit von G_1 und G_2 unter M ist gegeben durch

$$BSim_M(G_1, G_2) = \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} Sim_{funk}(p, M(p)) \in [0, 1].$$

Dieses Maß erlaubt nun, die Ähnlichkeit von Aktivitätenbeschreibungen, d. h. der funktionalen Perspektive, unter einer M:N-Abbildung gemäß Definition 3.3 zu bestimmen. Mit den Methoden aus der Literatur, die weiter oben aufgeführt sind, ist das nicht direkt möglich.

4.2 Ressourcenbasierter Abgleich

Als Ressourcen eines Prozessmodells werden, wie schon in Abschnitt 3.2.2 bei der Definition des multiperspektivischen Prozessmodells erwähnt, in erster Linie menschliche Ressourcen, also die Agenten bzw. Ausführenden des Prozesses, sowie nichtmenschliche Ressourcen, also beispielsweise Werkzeuge, Software und Services, verstanden (Dumas et al., 2013). Auch Datenobjekte wie Dokumente, Vorlagen usw. werden im Folgenden im Begriff der Ressourcen eingeschlossen. Ressourcen haben die Eigenschaft, dass sie einzelnen Aktivitäten zugeordnet werden können, wobei zum Modellierzeitpunkt bereits die Mengen der zur Verfügung stehenden Ressourcen bzw. ihrer eindeutigen Bezeichner bekannt sind, z. B. „Vorlage Dienstreiseantrag“ (eindeutiges Datenobjekt) oder „Leiter Abteilung A“ (eindeutige Rollenbezeichnung). Diese Mengen sind zudem endlich.

Von Weidlich et al. (2010a) werden die den Aktivitäten zugewiesenen Agenten/Rollen sowie die ausgehenden Datenobjekte im Ähnlichkeitsabgleich insoweit berücksichtigt, als dass die Bezeichnungen als Label (ähnlich den Aktivitätsbeschreibungen) aufgefasst werden und auf diesen labelbasierte Ähnlichkeitsberechnungen durchgeführt werden. Für eine Menge an Aktivitäten werden die entsprechenden Label konkateniert und mit Stemmingmethoden (siehe Abschnitt 2.2.2.2) normalisiert. Dies sind die in Abbildung 3.7 eingezeichneten, vorhandenen Methoden beim 1:N-Abgleich der organisatorischen und datenorientierten Perspektive. Eine Ausweitung dieses Ansatzes auf M:N-Abgleiche ist einfach zu bewerkstelligen, da es sich im Grunde genommen um Labelähnlichkeit handelt, die in Abschnitt 4.1 für M:N-Abbildungen erklärt ist. Die Organisationsstruktur, die hinter den im Modell genannten Personen oder Rollen liegt, wird bei dieser Methode beispielsweise komplett ignoriert. Syntaktisch ähnliche Rollennamen müssen aber nichts zwangsläufig auch auf ähnliche Rollen hinweisen. Gleiches gilt für Datenobjekte, die z. B. über eine Nummer angesprochen werden („Vorlage_73B“). Gelten zudem perspektivenübergreifende Regeln zur Bestimmung der möglichen Ressourcen, so ist der von Weidlich et al. (2010a) vorgeschlagene Ansatz nicht mehr sinnvoll durchführbar, da dann auch textuelle Beschreibungen von Zuweisungsregeln mit abgeglichen werden.

In Abschnitt 4.2.1 werden zunächst Ähnlichkeitsmaße für die organisatorische Perspektive entwickelt. Abhängig von der zugrunde gelegten Organisationsstruktur kann hierbei zwischen mehreren Möglichkeiten unterschieden werden. In Abschnitt 4.2.2 wird eines der Maße für die organistorische Perspektive auf Datenobjekte übertragen. Die gleiche Übertragung wird in Abschnitt 4.2.3 für die operationale Perspektive durchgeführt. Anschließend findet sich in Abschnitt 4.2.4 eine erste, qualitative Bewertung der Ähnlichkeitsmessmethoden der organisatorischen und datenorientierten Perspektiven anhand von in der Literatur genannten Kriterien. Die operationale Perspektive ist dort nicht genannt.

4.2.1 Abgleich der organisatorischen Perspektive

Um die organisatorische Perspektive von Aktivitätenmengen zu vergleichen, werden im Folgenden verschiedene Ansätze gezeigt, die teilweise bereits in den Arbeiten der Autorin und ihrer Co-Autoren (Baumann et al., 2014, 2015b) vorgestellt werden. Zunächst einmal muss die Grundlage der organisatorischen Perspektive geklärt werden. Werden Prozesse innerhalb eines Unternehmens betrachtet, so kann davon ausgegangen werden, dass das zugrunde liegende organisatorische Modell für alle Prozessmodelle dasselbe ist, d. h. insbesondere, dass Rollennamen und Mitarbeiterkennungen in zwei zu vergleichenden Prozessmodellen direkt miteinander vergleichbar sind. Handelt es sich bei den zu vergleichenden Prozessmodellen um Modelle aus verschiedenen Organisationen, so muss im Vorfeld ein (manueller) Abgleich der Ressourcen bzw. des Ressourcenmodells durchgeführt werden. Es sei im Folgenden $\mathcal{A}$ die gemeinsame, bereits abgeglichene Menge der organisatorischen Einheiten für alle abzugleichenden Prozessmodelle.

Von Bussler (1998) wird die Organisationsverwaltung, also die Verwaltung der Organisationsstrukturen innerhalb eines Unternehmens, ausführlich beschrieben und diskutiert. Die drei Teile, aus denen die Organisationsverwaltung bei Bussler (1998) besteht, sind die Organisationsstrukturverwaltung, die Zuweisungsregelverwaltung und die Synchronisationsregelverwaltung, wobei für den Ähnlichkeitsabgleich lediglich der erste Teil, die Strukturverwaltung, von Bedeutung ist, da die beiden anderen Teile die Ausführung der Prozesse betreffen. In der Organisationsstrukturverwaltung werden Organisationsstrukturen festgelegt, die Gruppen und Rollen miteinander in Beziehung setzen. Außerdem können über Auswertungsfunktionalitäten die tatsächlichen Agenten, die hinter den jeweiligen Gruppen und Rollen stehen, angefragt werden. Die Organisationsstruktur, die Zuordnung von Objekten dieser Struktur zu Prozessaufgaben sowie die Auswertung der Gruppen und Rollen werden für einen Ähnlichkeitsabgleich der organisatorischen Perspektive benötigt. Die Zuweisungsregelverwaltung legt zur Laufzeit fest, wie die zu erledigenden Aufgaben an die einzelnen Personen verteilt werden, und die Synchronisationsregelverwaltung regelt den Zugriff von Personen auf das Prozessausführungssystem, um beispielsweise zu verhindern, dass eine Aufgabe unnötigerweise gleichzeitig von zwei Personen erledigt wird.

Im Nachfolgenden werden mehrere Möglichkeiten zur Ähnlichkeitsbestimmung von zugewiesenen Rollen und Agenten vorgestellt. Abschnitt 4.2.1.1 setzt hierbei eine hierarchische Organisationsstruktur voraus, während die Methoden aus Abschnitt 4.2.1.2 eine solche nicht benötigen.

4.2.1.1 Ähnlichkeit für eine hierarchische Organisationsstruktur

Die erste Möglichkeit zur Ähnlichkeitsberechnung wird von Baumann et al. (2014) vorgestellt und setzt eine hierarchisch aufgebaute, baumartige Organisationsstruktur (vgl. Bertino et al.,

1999), wie sie z. B. in Abbildung 4.1 gegeben ist, voraus. Hierbei bezeichnen die Buchstaben A bis F konkrete Gruppen bzw. Rollen, wobei beispielsweise die Gruppen C und E auf gleicher Ebene liegen, beide jedoch der Gruppe D unterstellt sind. Die Gruppen D und B befinden sich auf gleicher Ebene und sind F unterstellt. Indirekt sind auch A, C und E der Gruppe F unterstellt. A ist direkt B unterstellt.

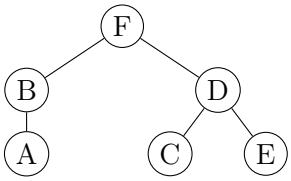


Abbildung 4.1: Hierarchisch aufgebaute Organisationsstruktur

Das im Folgenden beschriebene Ähnlichkeitsmaß betrachtet den Abstand der organisatorischen Einheiten im hierarchischen Modell. Der Abstand zweier Knoten in einem hierarchischen Modell kann zum einen in der minimalen Anzahl an Kanten, die zwischen den beiden Knoten liegen, gemessen werden, zum anderen auch in der Differenz der Höhe, auf der sich die Knoten im Baum befinden. Es bezeichne $\tilde{k}(\cdot,\cdot)$ die minimale Anzahl an Kanten, die von einem Knoten zu einem anderen passiert werden müssen, und $\tilde{e}(\cdot,\cdot)$ den Höhenunterschied der beiden Knoten. Um aus diesen beiden Distanzmaßen Ähnlichkeitsmaße zu generieren, wird dieselbe Umrechnung wie in Gleichung (2.2) verwendet. Die Kantenähnlichkeit $ksim$ zwischen zwei Knoten n_1 und n_2 im hierarchischen Organisationsmodell ist somit

$$ksim(n_1, n_2) = \frac{1}{\tilde{k}(n_1, n_2) + 1} \in (0, 1]$$

und die Ebenenähnlichkeit $esim$ der beiden Knoten

$$esim(n_1, n_2) = \frac{1}{\tilde{e}(n_1, n_2) + 1} \in (0, 1].$$

Die Kantenähnlichkeit ist nur dann 1, wenn $n_1 = n_2$ gilt, während die Ebenenähnlichkeit auch 1 sein kann für $n_1 \neq n_2$. Im Beispiel aus Abbildung 4.1 ist $ksim(C, E) = 1/(2+1) = 1/3$ und $esim(C, E) = 1/(0+1) = 1$. Ein Vergleich der beiden Gruppen A und E liefert dahingegen $ksim(A, E) = 1/(1+4) = 1/5$ und $esim(A, E) = 1/(0+1) = 1$.

Die Kanten- und Ebenenähnlichkeit wird mittels Gewichtung zu einem Ähnlichkeitswert verrechnet: Für $\alpha \in [0, 1]$ ist die Ähnlichkeit zweier organisatorischer Gruppen n_1 und n_2 definiert als

$$hsim(n_1, n_2) = \alpha \cdot ksim(n_1, n_2) + (1 - \alpha) \cdot esim(n_1, n_2) \in (0, 1].$$

Mit α kann festgelegt werden, ob eine grundsätzliche Nähe der organisatorischen Einheiten im Baum oder die Gleichheit der Ebene stärker bei der Ähnlichkeitsberechnung berücksichtigt werden soll. Sollen beispielsweise Zugehörigkeiten zur selben Abteilung stärker ins Ähnlichkeitsmaß mit eingehen, muss $ksim$ stärker gewichtet werden, sollen dagegen gleiche Positionen, wie z. B. Einkauf Abteilung A und Einkauf Abteilung B, eher beachtet werden, muss $esim$ stärker gewichtet werden.

Mit Hilfe von $hsim$, das für zwei einzelne organisatorische Einheiten definiert ist, wird nun ein Ähnlichkeitsmaß für zwei Mengen an Aktivitäten definiert, denen jeweils eine Menge an organisatorischen Einheiten durch das jeweilige Prozessmodell zugeordnet ist. Seien

$G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle, auf denen die Abbildung M die Partitionen $P_1 \subseteq \mathcal{P}(A_1)$ und $P_2 \subseteq \mathcal{P}(A_2)$ induziert, $M : P_1 \to P_2$. Zunächst wird für zwei zu vergleichende Mengen an Aktivitäten die Menge aller diesen Aktivitäten zugewiesener Agenten bestimmt, wobei Agenten in diesem Fall die Gruppen bzw. Rollen aus dem hierarchischen Modell sind. Für $p_i \in P_j$ ist die Menge aller zugewiesenen Agenten $\mathcal{A}_{p_i}$ die Vereinigung der Agenten der einzelnen Knoten, aus denen p_i besteht, d. h.

$$\mathcal{A}_{p_i} = \{m \mid \exists n \in p_i \in P_j : m \in \lambda_{j,2}(n)\}.$$

Es wird hier angenommen, dass jeder Aktivität mindestens ein Agent zugeordnet ist; es ist also immer $\mathcal{A}_{p_i} \neq \emptyset$. Werden nun p_1 und p_2 verglichen, so wird über die Ähnlichkeit von jedem Agenten aus p_1 zu jedem Agenten aus p_2 gemittelt. Die Ähnlichkeit von einem Agenten aus p_1 zu einem Agenten aus p_2 errechnet sich über die Funktion $hsim$.

Definition 4.3 (Ähnlichkeit für eine hierarchische Organisationsstruktur)**.** Für Prozessmodelle G_1 und G_2 mit Abbildung $M : P_1 \to P_2$ und einer hierarchischen Organisationsstruktur der Elemente aus $\mathcal{A}$ ist die organisatorische Ähnlichkeit Sim_{hch} definiert durch

$$Sim_{hch}(p_1, p_2) = \frac{\sum_{m_1 \in \mathcal{A}_{p_1}, m_2 \in \mathcal{A}_{p_2}} hsim(m_1, m_2)}{|\mathcal{A}_{p_1}| \cdot |\mathcal{A}_{p_2}|} \in [0, 1].$$

Beispiel 4.1. Für die Aktivitäten aus Abbildung 4.2 mit den zugeordneten Agenten und der angegebenen Abbildung berechnet sich die organisatorische Ähnlichkeit für die hierarchische Organisationsstruktur aus Abbildung 4.1 wie folgt. Zunächst werden die Agenten der Aktivitätsmengen bestimmt:

- $\mathcal{A}_{p_1} = \{A, B\}$
- $\mathcal{A}_{p_2} = \{A, B\}$

Anschließend wird für jedes Agentenpaar die Ähnlichkeit anhand des hierarchischen Modells bestimmt. Für das Beispiel wird $\alpha = 1/2$ gewählt.

- $hsim(A, A) = \frac{1}{2} \cdot ksim(A, A) + \frac{1}{2} \cdot esim(A, A) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 1$
- $hsim(A, B) = \frac{1}{2} \cdot ksim(A, B) + \frac{1}{2} \cdot esim(A, B) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$
- $hsim(B, A) = \frac{1}{2} \cdot ksim(B, A) + \frac{1}{2} \cdot esim(B, A) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$
- $hsim(B, B) = \frac{1}{2} \cdot ksim(B, B) + \frac{1}{2} \cdot esim(B, B) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 1$

Die hierarchische Ähnlichkeit ist dann der Mittelwert über die einzelnen Ähnlichkeiten der Agentenpaare:

$$Sim_{hch}(p_1, p_2) = \frac{1}{4} \cdot \left(1 + \frac{1}{2} + \frac{1}{2} + 1\right) = \frac{3}{4} = 0,75$$

Wie am Beispiel 4.1 ersichtlich, ist das Ähnlichkeitsmaß sim_{hch} so konstruiert, dass auch wenn p_1 und p_2 die gleiche Menge an zugewiesenen Agenten haben, ihre Ähnlichkeit nicht unbedingt 1 sein muss. Das ist dann der Fall, sobald entweder $|\mathcal{A}_{p_1}| > 1$ oder $|\mathcal{A}_{p_2}| > 1$ ist. Dies ist insofern beabsichtigt, als dass Abbildungen, die Aktivitäten mit unterschiedlichen Agenten kombinieren, durch einen geringeren Wert von sim_{hch} bestraft werden. Oder

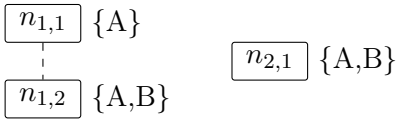


Abbildung 4.2: Beispielzuordnung mit angegebener organisatorischer Perspektive: $p_1 = \{n_{1,1}, n_{1,2}\}, p_2 = \{n_{2,1}\}, M(p_1) = p_2$; $\lambda_{1,2}(n_{1,1}) = \{A\}, \lambda_{1,2}(n_{1,2}) = \{A, B\}, \lambda_{2,2}(n_{2,1}) = \{A, B\}$

umgekehrt: Es werden solche Abbildungen bevorzugt, die, bezogen auf die Agenten, gleichartige Aktivitäten kombinieren. Sind einer Aktivität allerdings mehrere Agenten zugeordnet, so führt dies ebenfalls notgedrungen zu einer geringeren Ähnlichkeit. Somit erfüllt Sim_{hch} nicht in jedem Fall die Definition eines Ähnlichkeitsmaßes (Definition 2.1). Eine Modifikation, die zu jedem Agenten den ähnlichsten Agenten der anderen Aktivitätenmenge sucht (wie beispielsweise bei der Bag of Words-Ähnlichkeit) ist ebenfalls möglich. Dies erhöht zwar den Berechnungsaufwand, das Ähnlichkeitsmaß ist aber tatsächlich ein Maß:

$$Sim_{hchmax}(p_1, p_2) = \frac{\sum\limits_{m_1 \in \mathcal{A}_{p_1}} \max\limits_{m_2 \in \mathcal{A}_{p_2}} hsim(m_1, m_2) + \sum\limits_{m_2 \in \mathcal{A}_{p_2}} \max\limits_{m_1 \in \mathcal{A}_{p_1}} hsim(m_1, m_2)}{|\mathcal{A}_{p_1}| \cdot |\mathcal{A}_{p_2}|}$$

Eine Bestrafung inhomogener Agentenmengen findet dann allerdings nicht statt, weswegen Sim_{hchmax} als Ähnlichkeitsmaß nicht unbedingt verwendet werden sollte.

4.2.1.2 Ähnlichkeit für eine beliebige Organisationsstruktur

Ist die Organisationsstruktur nicht als hierarchischer Baum gegeben, sondern in einer beliebigen Form, kann die in Abschnitt 4.2.1.1 gegebene Ähnlichkeitsberechnung nicht angewendet werden. Stattdessen wird ein Ähnlichkeitsmaß basierend auf dem Jaccard-Koeffizienten definiert. Dieses Vorgehen wird von Baumann et al. (2015b) vorgestellt. Hierfür bestimmt man für jedes $p_i \in P_j$ wieder die Menge der zuständigen Agenten $\mathcal{A}_{p_i}$:

$$\mathcal{A}_{p_i} = \{m \mid \exists n \in p_i \in P_j : m \in \lambda_{j,2}(n)\}. \tag{4.1}$$

Es wird anschließend der Anteil gleicher Agenten in den zu vergleichenden Mengen an Aktivitäten berechnet. Dieser Wert ist die Ähnlichkeit der beiden Aktivitätenmengen. Grundsätzlich kann diese Form des Abgleichs analog zur Jaccard-Ähnlichkeit auf Aktivitätenbeschreibungen (siehe Abschnitt 2.2.2.3) gesehen werden, mit dem Unterschied, dass hier keine Menge an (beliebigen) Wörtern sondern eine Menge an (festen) Bezeichnern verwendet wird.

Definition 4.4 (Ähnlichkeit für beliebige Organisationsstruktur (alle Agenten))**.** Für Prozessmodelle G_1 und G_2 mit Abbildung $M : P_1 \to P_2$ und einer beliebigen Organisationsstruktur der Elemente aus $\mathcal{A}$ ist die organisatorische Ähnlichkeit Sim_{org} definiert durch

$$Sim_{org}(p_1, p_2) = \frac{|\mathcal{A}_{p_1} \cap \mathcal{A}_{p_2}|}{|\mathcal{A}_{p_1} \cup \mathcal{A}_{p_2}|} \in [0, 1].$$

Beispiel 4.2. Für die Aktivitäten aus Abbildung 4.2 mit den zugeordneten Agenten und der angegebenen Abbildung berechnet sich die organisatorische Ähnlichkeit für eine beliebige Organisationsstruktur auf Grundlage der Vereinigung aller Agenten wie folgt. Wie in Beispiel 4.1

ist $\mathcal{A}_{p_1} = \{A, B\}$ und $\mathcal{A}_{p_2} = \{A, B\}$. Es folgt

$$Sim_{org}(p_1, p_2) = \frac{|\mathcal{A}_{p_1} \cap \mathcal{A}_{p_2}|}{|\mathcal{A}_{p_1} \cup \mathcal{A}_{p_2}|} = \frac{|\{A, B\} \cap \{A, B\}|}{|\{A, B\} \cup \{A, B\}|} = \frac{2}{2} = 1.$$

Das Ähnlichkeitsmaß sim_{org} hat im Gegensatz zu sim_{hch} nicht die zwingende Eigenschaft, inhomogene Mengen an Aktivitäten über einen geringeren Ähnlichkeitswert zu bestrafen, wie Beispiel 4.2 zeigt. Dies liegt daran, dass in $\mathcal{A}_{p_i}$ alle Agenten versammelt sind, die für mindestens eine der in p_i enthaltenen Aktivitäten zuständig sind. Über die Vereinigung in $\mathcal{A}_{p_i}$ wird dann suggeriert, dass alle Agenten für alle in p_i enthaltenen Aktivitäten zuständig und vor allem auch befugt sind. Also dass beispielsweise Agent B auch Aktivität $n_{1,1}$ ausführen kann. Dies kann jedoch in gewissen Fällen problematisch sein, z. B. wenn es um explizite Aufgabentrennung (*separation of duties*) geht.[1] Um also eine, bezogen auf die Agenten, homogene Aktivitätenmenge zu bevorzugen, sollten in der Agentenmenge von p_i nur solche Agenten enthalten sein, die für alle Aktivitäten aus p_i zuständig sind. Diese Agentenmenge ist

$$\mathcal{A}'_{p_i} = \{m \mid \forall n \in p_i \in P_j : m \in \lambda_{j,2}(n)\}. \tag{4.2}$$

Auch auf Basis von $\mathcal{A}'_{p_i}$ kann ein Jaccard-Koeffizient berechnet werden.

Definition 4.5 (Ähnlichkeit für beliebige Organisationsstruktur (gemeinsame Agenten))**.** Für Prozessmodelle G_1 und G_2 mit Abbildung $M : P_1 \to P_2$ und einer beliebigen Organisationsstruktur der Elemente aus $\mathcal{A}$ ist die organisatorische Ähnlichkeit $Sim_{org'}$ definiert durch

$$Sim_{org'}(p_1, p_2) = \frac{|\mathcal{A}'_{p_1} \cap \mathcal{A}'_{p_2}|}{|\mathcal{A}'_{p_1} \cup \mathcal{A}'_{p_2}|} \in [0, 1]$$

für $|\mathcal{A}'_{p_1} \cup \mathcal{A}'_{p_2}| > 0$. Falls $|\mathcal{A}'_{p_1} \cup \mathcal{A}'_{p_2}| = 0$, dann ist $sim_{org'}(p_1, p_2) = 0$.

Die Fallunterscheidung in Definition 4.5 muss getroffen werden für den Fall, dass weder in p_1 noch in p_2 gemeinsame Agenten vorhanden sind, also dass sowohl $\mathcal{A}'_{p_1} = \emptyset$ als auch $\mathcal{A}'_{p_2} = \emptyset$.

Beispiel 4.3. Für die Aktivitäten aus Abbildung 4.2 mit den zugeordneten Agenten und der angegebenen Abbildung berechnet sich die organisatorische Ähnlichkeit für eine beliebige Organisationsstruktur auf Grundlage des Schnitts aller Agenten wie folgt. Es ist $\mathcal{A}'_{p_1} = \{A\}$ und $\mathcal{A}'_{p_2} = \{A, B\}$. Es folgt damit

$$Sim_{org'}(p_1, p_2) = \frac{|\mathcal{A}'_{p_1} \cap \mathcal{A}'_{p_2}|}{|\mathcal{A}'_{p_1} \cup \mathcal{A}'_{p_2}|} = \frac{|\{A\} \cap \{A, B\}|}{|\{A\} \cup \{A, B\}|} = \frac{|\{A\}|}{|\{A, B\}|} = \frac{1}{2}.$$

Bei Verwendung des Ähnlichkeitsmaßes $Sim_{org'}$ werden nun allerdings mögliche Agenten unterschlagen, wie das Beispiel 4.4 zeigt, und somit Unterschiede in den zu vergleichenden Aktivitätenmengen nicht berücksichtigt.

[1] Die Regeln an sich, also beispielsweise die *separation of duties*-Regel, werden auf diese Weise nicht direkt berücksichtigt. Sie sind vielmehr in den bereits vorgenommenen, deterministischen Zuweisungen von bestimmten Rollen oder Agenten zu Aktivitäten enthalten. Werden Regeln für die Angabe der organisatorischen Perspektive verwendet, kann das angegebene Ähnlichkeitmaß nicht sinnvoll angewendet werden.

Beispiel 4.4. Für die Aktivitäten aus Abbildung 4.3 mit den zugeordneten Agenten und der angegebenen Abbildung berechnet sich die organisatorische Ähnlichkeit für eine beliebige Organisationsstruktur auf Grundlage des Schnitts aller Agenten wie folgt. Es ist $\mathcal{A}'_{p_1} = \{A\}$ und $\mathcal{A}'_{p_2} = \{A\}$. Es folgt damit

$$Sim_{org'}(p_1, p_2) = \frac{|\{A\} \cap \{A\}|}{|\{A\} \cup \{A\}|} = 1.$$

Das Vorkommen der Agenten B, C, D und E in Aktivität $n_{1,2}$ verringert den Ähnlichkeitswert nicht.

Abbildung 4.3: Beispielzuordnung mit angegebener organisatorischer Perspektive: $p_1 = \{n_{1,1}, n_{1,2}\}, p_2 = \{n_{2,1}\}, M(p_1) = p_2$; $\lambda_{1,2}(n_{1,1}) = \{A\}, \lambda_{1,2}(n_{1,2}) = \{A, B, C, D, E\}$, $\lambda_{2,2}(n_{2,1}) = \{A\}$.

Um die Vorteile beider Ähnlichkeitsmaße Sim_{org} und $Sim_{org'}$ zu vereinen, kann ein hybrider Ansatz als Ähnlichkeit gewählt werden, der kein Jaccard-Koeffizient ist, da die Grundmengen nicht mehr dieselben sind. Dieser Ansatz misst den Anteil der Agenten, die für jede Aktivität der zur vergleichenden Aktivitätenmengen p_1 und p_2 zuständig sind, bezogen auf alle vorkommenden Agenten der Aktivitäten aus p_1 und p_2. Er nutzt aus, dass $\mathcal{A}'_{p_i} \subseteq \mathcal{A}_{p_i}$.

Definition 4.6 (Ähnlichkeit für beliebige Organisationsstruktur (hybrider Ansatz))**.** Für Prozessmodelle G_1 und G_2 mit Abbildung $M : P_1 \to P_2$ und einer beliebigen Organisationsstruktur der Elemente aus $\mathcal{A}$ ist die organisatorische Ähnlichkeit $Sim_{org''}$ definiert durch

$$Sim_{org''}(p_1, p_2) = \frac{|\mathcal{A}'_{p_1} \cap \mathcal{A}'_{p_2}|}{|\mathcal{A}_{p_1} \cup \mathcal{A}_{p_2}|} \in [0, 1].$$

Beispiel 4.5. Für die Aktivitäten aus Abbildung 4.3 mit den zugeordneten Agenten und der angegebenen Abbildung berechnet sich die organisatorische Ähnlichkeit für eine beliebige Organisationsstruktur auf Grundlage des hybriden Ansatzes wie folgt. Es sind $\mathcal{A}_{p_1} = \{A, B, C, D, E\}$, $\mathcal{A}_{p_2} = \{A\}$, $\mathcal{A}'_{p_1} = \{A\}$ und $\mathcal{A}'_{p_2} = \{A\}$. Es folgt damit

$$Sim_{org''}(p_1, p_2) = \frac{|\{A\} \cap \{A\}|}{|\{A, B, C, D, E\} \cup \{A\}|} = \frac{1}{5}.$$

Das Vorkommen der Agenten B, C, D und E in Aktivität $n_{1,2}$ beeinflusst nun den Ähnlichkeitswert. Zum Vergleich: Die Ähnlichkeit Sim_{org} für die Aktivitäten aus Abbildung 4.3 wäre

$$Sim_{org}(p_1, p_2) = \frac{|\{A, B, C, D, E\} \cap \{A\}|}{|\{A, B, C, D, E\} \cup \{A\}|} = \frac{1}{5} = Sim_{org''}(p_1, p_2).$$

Für den Modellausschnitt in Abbildung 4.2 ergibt sich für den hybriden Ansatz ein Wert von

$$Sim_{org''}(p_1, p_2) = \frac{|\{A\} \cap \{A, B\}|}{|\{A, B\} \cup \{A, B\}|} = \frac{1}{2} = Sim_{org'}(p_1, p_2).$$

Der hybride Ansatz $Sim_{org''}$ ist wegen $\mathcal{A}'_{p_i} \subseteq \mathcal{A}_{p_i}$ strenger als die beiden anderen Ansätze, d. h. $Sim_{org''} \leq Sim_{org}$ und $Sim_{org''} \leq Sim_{org'}$.

Es kann also gesagt werden, dass wenn eine hierarchische Organisationsstruktur vorliegt und nach Möglichkeit nicht mehrere Rollen/Agenten einzelnen Aktivitäten zugeordnet sind, die Methode Sim_{hch} zur Berechnung der Ähnlichkeit der angegebenen Agenten verwendet werden kann, da diese die Informationen über die Hierarchie mit berücksichtigt. Liegt keine hierarchische Organisationsstruktur vor, so sollte Methode $Sim_{org''}$ verwendet werden, da $Sim_{org''}$ ohne Einschränkung verwendet werden kann.

Die Ähnlichkeit zweier Prozessmodelle bezüglich der organisatorischen Perspektive ist dann:

Definition 4.7 (Agentenähnlichkeit)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(A_1)$ und $P_2 \subseteq \mathcal{P}(A_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Die Agentenähnlichkeit von G_1 und G_2 unter M ist gegeben durch

$$ASim_M(G_1, G_2) = \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} Sim^*(p, M(p)) \in [0, 1]$$

wobei für Sim^* entweder Sim_{hch} oder $Sim_{org''}$ eingesetzt wird.

Es sei an dieser Stelle nochmals erwähnt, dass Abbildungen, die alle Aktivitäten löschen, nicht erlaubt sind. Sind in einem Prozessmodell Agenten über indirekte Anweisungen den Aktivitäten zugeordnet, beispielsweise bei einer perspektivenübergreifenden Modellierung, wie sie in Abschnitt 3.2.4 angesprochen ist, die nur zur Laufzeit aufgelöst werden kann, so muss bei einem Abgleich der organisatorischen Perspektive in diesem Fall auf die den Rollen und Gruppen zugewiesenen realen Personen, die über eine Mitarbeiterkennung voneinander unterschieden werden können, zurückgegriffen werden. Von Cabanillas (2016) werden die Personen, die sich aus dieser Auflösung ergeben, als *potential participants* bezeichnet. Falls es beispielsweise heißt, Äktivität B muss vom Chef derjenigen Person ausgeführt werden, die Aktivität A ausgeführt hat (außer diese war selbst der oberste Chef), so wird erst für Aktivität A die Menge aller möglichen Ausführenden bestimmt und von diesen jeweils über die Auswertungsfunktionalitäten der Organisationsstrukturverwaltung die Chefs bestimmt. In den Agentenmengen der beiden Aktivitäten sind dann jeweils die Mitarbeiterkennungen aller in Frage kommenden Personen vereint. Das gleiche Prinzip gilt ebenfalls, wenn keine Eins-zu-eins-Entsprechungen in den zugewiesenen Rollen oder Gruppen gefunden werden können, z. B.: „CR Team 1“ ist nicht gleich mit „CR Abteilung“, wenn es im ersten Fall noch ein „CR Team 2“ gibt, aber auch nicht komplett verschieden, wenn sich die CR-Abteilung aus „CR Team 1“ und „CR Team 2“ zusammensetzt. Auch hier sollte auf die den Gruppennamen hinterlegten konkreten Personen zurückgegriffen werden, für die dann, bei etwa gleicher Teamstärke, eine Ähnlichkeit von ca. 0,5 zwischen „CR Abteilung“ und „CR Team 1“ herauskommen sollte. Dieser Ähnlichkeitswert hängt aber von der aktuellen Zuteilung von Personen zu Rollen und Gruppen ab. Im Fall der nur teilweisen Entsprechung von bestimmten Rollen und Gruppen können auch die zugrunde liegenden Prozessmodelle unter Einbezug der Organisationsstrukturverwaltung dahingehend angepasst werden, dass alle Agentenzuweisungen auf die kleinste Granularität heruntergebrochen werden. Im Beispiel von oben: Statt im einen Prozessmodell „CR Abteilung“ der Aktivität als Agent zuzuweisen, kann, wenn festgestellt wird, dass im anderen Modell die Zuweisungen eine Ebene tiefer liegen, eine Ersetzung von „CR Abteilung“

durch die beiden Gruppen „CR Team 1“ und „CR Team 2“ erfolgen. Notfalls kann die kleinste Granularität wieder die einzelne, konkrete Person sein. Eine detailliertere Betrachtung der organisatorischen Perspektive wird auch in Abschnitt 6.2 angesprochen, in dem auf zukünftige Forschungsarbeiten eingegangen wird.

Nachdem in diesem Abschnitt mehrere Möglichkeiten für eine Ähnlichkeitsberechnung der organisatorischen Perspektive definiert sind, folgt in Abschnitt 4.2.2 die datenorientierte Perspektive. Hierfür wird eine der Methoden der Agentenähnlichkeit auf Datenobjekte übertragen, wobei die Übertragbarkeit auf Basis gleicher Ressourcenstruktur (Abschnitt 3.5.2) ausgenutzt wird.

4.2.2 Abgleich der datenorientierten Perspektive

Der Abgleich der datenorientierten Perspektive kann grundsätzlich wie der der organisatorischen Perspektive erfolgen, vorausgesetzt, die in Abschnitt 3.5.2 genannte und in Abbildung 3.9 veranschaulichte Übertragbarkeit ist möglich. Wie in der Prozessmodelldefinition 3.1 angegeben, können hier nur Bezeichner für Datenobjekte berücksichtigt werden, da konkrete Werte, beispielsweise in einem Formular eingetragene Zahlen, erst zur Laufzeit verfügbar sind, nicht zum Modellierzeitpunkt, zu dessen Stand der Abgleich erfolgen soll. Künzle und Reichert (2009) sprechen hier von *object types* im Gegensatz zu *object instances* (Instanzen von *object types* zur Laufzeit).

Für das Maß Sim_{hch} wird eine hierarchische Organisationsstruktur der Ressourcen vorausgesetzt. Diese kann für Datenobjekte in den meisten Fällen jedoch nicht angenommen werden, weshalb eine Übertragung von Sim_{hch} auf die Datenperspektive im Allgemeinen nicht möglich ist.

Ein weiterer Unterschied zwischen der organisatorischen und der datenorientierten Perspektive ist der, dass bei der Datenperspektive keine Homogenität der Dokumentenmengen in den zusammengefassten Aktivitäten angestrebt werden muss. Werden zwei Aktivitäten des ursprünglichen Modells in einer Modellvariante zusammengefasst, so werden einfach alle in den zwei einzelnen Aktivitäten verwendeten Datenobjekte in der zusammengefassten Aktivität im neuen Modell ebenfalls zusammengefasst. Es ist also möglich, das Maß Sim_{org}, das für die organisatorische Perspektive weniger gut geeignet ist, für den Abgleich der Datenperspektive heranzuziehen. Da bei der Definition des multiperspektivischen Prozessmodells (Definition 3.1) nicht zwischen ein- und ausgehenden Datenobjekten unterschieden wird, kann bei der Definition eines Ähnlichkeitsmaßes diese Unterscheidung ebenfalls nicht gemacht werden. Allerdings ist eine solche Unterscheidung leicht zu bewerkstelligen, da die Mengen der ein- und ausgehenden Datenobjekte wie die Menge aller Datenobjekte behandelt werden können. Lediglich aus Gründen der kompakteren Darstellbarkeit wird diese Unterscheidung hier (bzw. in der Definition des multiperspektivischen Modells) nicht gemacht. Eine Unterscheidung lässt tendenziell genauere Ergebnisse bei einem Abgleich erwarten, da sie die Funktion eines Datenobjekts innerhalb einer Aktivität berücksichtigt, was durch die reine Verwendung eines Datenobjektes bei einer Aktivität nicht geschieht.

Definition 4.8 (Ähnlichkeit der zugewiesenen Datenobjekte (alle Datenobjekte))**.** Für Prozessmodelle G_1 und G_2 mit Abbildung $M : P_1 \to P_2$ und einer gegebenen Menge an Datenobjekten $\mathcal{D}$ ist die datenorientierte Ähnlichkeit Sim_{dat} definiert durch

$$Sim_{dat}(p_1, p_2) = \frac{|\mathcal{D}_{p_1} \cap \mathcal{D}_{p_2}|}{|\mathcal{D}_{p_1} \cup \mathcal{D}_{p_2}|} \in [0, 1],$$

falls $|\mathcal{D}_{p_1} \cup \mathcal{D}_{p_2}| > 0$, wobei

$$\mathcal{D}_{p_i} = \{d \mid \exists n \in p_i \in P_j : d \in \lambda_{j,3}(n)\}.$$

Falls $|\mathcal{D}_{p_1} \cup \mathcal{D}_{p_2}| = 0$, so ist $Sim_{dat}(p_1, p_2) = 1$.

Eine Übertragung von $Sim_{org'}$ ist zwar möglich, jedoch wenig sinnvoll. Eine Zuweisung von einer Menge an Datenobjekten zu einer Menge an Aktivitäten, wobei in der Menge der Datenobjekte nur die Objekte enthalten sind, die in allen einzelnen Aktivitäten verwendet werden, unterschlägt möglicherweise viele weitere verwendete Datenobjekte. Gleiches gilt für eine Übertragung von $Sim_{org''}$. Auch hier wird im Zähler der Berechnungsformel (siehe Definition 4.6) $\mathcal{A}'_{p_i}$ verwendet, also der Schnitt über alle Datenobjekte der in den zu einer Menge zusammengefassten Aktivitäten. Im Gegensatz zur organisatorischen Perspektive, die jeder Aktivität mindestens einen möglichen Agenten zuordnet, muss nicht jeder Aktivität im Modell ein Datenobjekt zugewiesen sein. Deswegen ist die Fallunterscheidung in Definition 4.8 zu treffen.

Die Ähnlichkeit zweier Prozessmodelle bezüglich der datenorientierten Perspektive ist dann:

Definition 4.9 (Datenähnlichkeit)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(A_1)$ und $P_2 \subseteq \mathcal{P}(A_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Die Datenähnlichkeit von G_1 und G_2 unter M ist gegeben durch

$$DSim_M(G_1, G_2) = \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} Sim_{dat}(p, M(p)) \in [0, 1].$$

4.2.3 Abgleich der operationalen Perspektive

Da die operationale Perspektive in der Angabe der den Aktivitäten zugewiesenen Services und Werkzeugen genau der datenorientierten Perspektive entspricht (siehe Definition 3.1), kann der Abgleich dieser Perspektive genau wie der der Datenperspektive erfolgen. Das Maß Sim_{dat} setzt keine Struktur der zugrunde liegenden Werkzeuge voraus, also ist eine Übertragung dieser Methode möglich.

Definition 4.10 (Ähnlichkeit der zugewiesenen Werkzeuge (alle Werkzeuge))**.** Für Prozessmodelle G_1 und G_2 mit Abbildung $M : P_1 \to P_2$ und einer gegebenen Menge an Services/Werkzeugen $\mathcal{S}$ ist die operationale Ähnlichkeit Sim_{opr} definiert durch

$$Sim_{opr}(p_1, p_2) = \frac{|\mathcal{S}_{p_1} \cap \mathcal{S}_{p_2}|}{|\mathcal{S}_{p_1} \cup \mathcal{S}_{p_2}|} \in [0, 1],$$

falls $|\mathcal{S}_{p_1} \cup \mathcal{S}_{p_2}| > 0$, wobei

$$\mathcal{S}_{p_i} = \{d \mid \exists n \in p_i \in P_j : d \in \lambda_{j,4}(n)\}.$$

Falls $|\mathcal{S}_{p_1} \cup \mathcal{S}_{p_2}| = 0$, so ist $Sim_{opr}(p_1, p_2) = 1$.

Die Ähnlichkeit zweier Prozessmodelle bezüglich der operationalen Perspektive ist dann:

Definition 4.11 (Serviceähnlichkeit)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(A_1)$ und $P_2 \subseteq \mathcal{P}(A_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Die Serviceähnlichkeit von G_1 und G_2 unter M ist gegeben durch

$$SSim_M(G_1, G_2) = \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} Sim_{opr}(p, M(p)) \in [0, 1].$$

Im Folgenden werden die Ähnlichkeitswerte der organisatorischen und der datenbasierten Perspektive hinsichtlich qualitativer Gütekriterien, die in der Literatur zu finden sind, untersucht.

4.2.4 Güte des ressourcenbasierten Abgleichs

Von Weidlich et al. (2008) werden einige Diskrepanzen von Prozessmodellen genannt, die in der datenorientierten und der organisatorischen Perspektive auftauchen können. Es wird nun überprüft, ob die Abweichungen durch die oben genannten Abgleichsmethoden ausgeglichen werden können, wenn sie auszugleichen sind, oder sich entsprechend im Ähnlichkeitswert niederschlagen. Es wird im Folgenden immer davon ausgegangen, dass die betrachteten Aktivitäten korrespondierende Aktivitäten sind, die eigentlich abgebildet werden müssten. Die genannten Abweichungen in der organisatorischen Perspektive, die von Weidlich et al. (2008) als „Ressourcenperspektive“ bezeichnet wird, sind die folgenden:

R.1 *Ressourcenfragmentierung: Es gibt keine Beziehung zwischen einzelnen Ressourcen, sondern zwischen Mengen an Ressourcen.*
Durch einen vorher festgelegten Abgleich der Ressourcen, der ggf. eine Aufsplittung von Gruppen in Untergruppen erfordert (gegen Ende von Abschnitt 4.2.1 angesprochen), kann die Ressourcenfragmentierung berücksichtigt werden. Alle Ähnlichkeitsmaße gehen immer von Mengen aus.

R.2 *Teilweise Ressourcenäquivalenz: Eine Ressource im einen Modell wird nur teilweise durch eine andere Ressource im zweiten Modell abgedeckt.*
Diese Abweichung kann, bei Abgleichen innerhalb eines Unternehmens, dadurch berücksichtigt werden, dass auf die konkreten Agenten, die bestimmte Rollen innehaben, zurückgegriffen wird (gegen Ende von Abschnitt 4.2.1 besprochen). Die nicht abgedeckten Agenten senken dann entsprechend den berechneten Ähnlichkeitswert zwischen den abgebildeten Aktivitäten.

R.3 *Nicht abgedeckte Ressourcen: Eine Ressource im einen Modell wird im zweiten Modell überhaupt nicht abgedeckt.*
Diese Abweichung stellt eine Unähnlichkeit der beiden Modelle bezüglich der Ressourcen der aufeinander abgebildeten Aktivitäten dar und wird entsprechend mit einer geringen Ähnlichkeit bewertet.

R.4 *Sich widersprechende Ressourcenzuweisungen: Einer Aktivität ist im einen Modell eine bestimmte Ressource zugewiesen, wobei die korrespondiere Aktivität im zweiten Modell einer anderen Ressource zugewiesen ist und diese Ressourcen widersprechen sich, das heißt, es ist bekannt, dass diese Ressourcen auf jeden Fall unterschiedlich sind.*
Ein Widerspruch kann mit den vorgestellten Methoden nicht direkt erkannt werden,

bezogen auf die Ressourcenähnlichkeit wird das Abbildungselement jedoch mit einer Ähnlichkeit von 0 bewertet, wobei dieser Ähnlichkeitswert als Ausschlusskriterium für die gesamte Abbildung verwendet werden kann (siehe Abschnitt 5.2.2.2). Entsprechen sich die Aktivitätenbeschreibungen, die Agenten aber nicht, wird ein „mittelmäßig guter“ gemittelter Ähnlichkeitswert als Ergebnis herauskommen. Dieser kann als Anlass genommen werden, um die Diskrepanzen zwischen den Prozessmodellen händisch aufzulösen.

R.5 *Zusätzliche Ressourcenzuweisungen: Einer Aktivität ist im einen Modell eine bestimmte Ressource zugewiesen, wobei die korrespondiere Aktivität im zweiten Modell einer anderen Ressource zugewiesen ist und diese Ressourcen sind unabhängig voneinander, das heißt es ist nicht bekannt, wie diese Ressourcen zusammenhängen.*
Werden die unabhängigen Ressourcen im Schritt des manuellen Abgleichs der Organisationsstrukturen nicht als gleich oder teilweise gleich eingestuft, so erhalten die Aktivitäten eine Agentenähnlichkeit von 0. Alles weitere gilt wie in R.4.

Für die datenorientierte Perspektive werden von Weidlich et al. (2008) ebenfalls fünf Punkte genannt, in denen sich eigentlich korrespondierende Aktivitäten bezüglich ihrer zugewiesenen Datenobjekte unterscheiden. Für D.1 bis D.3 gelten dieselben Aussagen wie für R.1 bis R.3.

D.1 *Datenobjektfragmentierung: Es gibt keine Beziehung zwischen einzelnen Datenobjekten, sondern zwischen Mengen an Datenobjekten.*
Wie in R.1: Durch einen vorher festgelegten Abgleich der Objekte, der ggf. eine Aufsplittung von Gruppen in Untergruppen erfordert, kann die Objektfragmentierung berücksichtigt werden. Das Ähnlichkeitsmaß geht immer von Mengen aus.

D.2 *Teilweise Datenobjektäquivalenz: Ein Objekt im einen Modell wird nur teilweise durch ein anderes Objekt im zweiten Modell abgedeckt.*
Wie in R.2 und D.1: Durch Aufsplittung der Objekte kann die Teiläquivalenz berücksichtigt werden. Die nicht abgedeckten Objekte der Aufspaltung senken dann entsprechend den berechneten Ähnlichkeitswert zwischen den abgebildeten Aktivitäten.

D.3 *Nicht abgedeckte Datenobjekte: Ein Datenobjekt im einen Modell wird im zweiten Modell überhaupt nicht abgedeckt.*
Wie R.3: Diese Abweichung stellt eine Unähnlichkeit der beiden Modelle bezüglich der Datenobjekte der aufeinander abgebildeten Aktivitäten dar und wird entsprechend mit einer geringen Ähnlichkeit bewertet.

D.4 *Unterschiedliche Anzahl an Datenobjektinstanzen: Ein Datenobjekt in einem Modell entspricht vielen Instanzen eines Datenobjekts (einer Liste) im anderen Modell.*
Auf Instanzbasis kann kein Ähnlichkeitsabgleich durchgeführt werden. Der Abgleich berücksichtigt nur die Bezeichner der Datenobjekte. Entspricht der Bezeichner der Liste im zweiten Modell dem einzelnen Datenobjekt im ersten Modell, so werden die Datenobjekte als ähnlich eingestuft. Dass mehrere Instanzen des Datenobjekts im zweiten Modell bei der Ausführung angelegt werden, wird nicht erkannt.

D.5 *Unterschiedlicher Datenzugriff: Es gibt einen Unterschied bezüglich des Datenzugriffs zwischen korrespondierenden Aktivitäten.*

Der Zugriff auf Datenobjekte wird beim Abgleich nicht berücksichtigt, da er bei der Definition des multiperspektivischen Prozessmodells (Definition 3.1) nicht auftaucht. Das heißt, es können insbesondere keine Diskrepanzen, aber auch keine Gemeinsamkeiten erkannt werden. Die einzige Unterscheidung, die bezüglich der Daten gemacht werden kann, ist die, zwischen eingehenden und ausgehenden Datenobjekten zu unterscheiden. Eventuell kann der Datenzugriff durch Angabe eines entsprechenden Werkzeugs/Services in der operationalen Perspektive angegeben werden.

Die genannten Diskrepanzen in der Struktur der Agenten und Datenobjekte (R.1-R.3 und D.1-D.3) können mit einem guten, manuellen Abgleich der Ressourcen im Vorab durch die mengenbasierte Ähnlichkeitsberechnung ausgeglichen werden. Fehlende Agenten bzw. Datenobjekte bedeuten einen Fehler im Modell und setzen sich entsprechend in der Ähnlichkeitsberechnung fort. Der Ähnlichkeitswert von nur teilweise übereinstimmenden Mengen ist < 1. Widersprüche in den Ressourcen können nicht direkt erkannt werden, da ein Widerspruch (bekanntermaßen unterschiedliche Ressourcen) wie eine Ungleichheit (unterschiedliche Ressourcen ohne Wissen, ob diese nicht doch zusammenhängen) behandelt wird. Hier müssten detailliertere Informationen der Ressourcenstruktur, wenn sie vorhanden sind, bei der Ähnlichhkeitsberechnung berücksichtigt werden. Instanzbezogene Diskrepanzen werden nicht erkannt, was dem Ansatz geschuldet ist, Prozesse auf Modellebene zu vergleichen. Des Weiteren können nur die Aspekte beim Ähnlichkeitsabgleich berücksichtigt werden, die im zugrunde liegenden Prozessmodell respräsentiert sind. Eine Beurteilung der jeweils fünf Kriterien ist in Tabelle 4.1 zusammengefasst.

Tabelle 4.1: Diskrepanzen in der organisatorischen und datenorientierten Perspektive und deren Berücksichtigung durch die gezeigten Ähnlichkeitsmaße (+ = kann berücksichtigt werden, o = kann teilweise berücksichtigt werden, - = kann nicht berücksichtigt werden).

	1	2	3	4	5
R.	+	+	+	o	o
D.	+	+	+	-	-

Es lässt sich erkennen, dass über die Hälfte der von Weidlich et al. (2008) genannten Diskrepanzen in den vorgestellten Ähnlichkeitsmaßen berücksichtigt werden. Die beiden teilweisen Berücksichtigungen in der organisatorischen Perspektive sind insoweit gut zu verschmerzen, da ein mittelmäßig guter Ähnlichkeitswert in den genannten Situationen gerade darauf hinweist, dass eventuell ein Modellierfehler vorliegt, da sich die Modelle nicht so ähnlich sind, wie sie möglicherweise sein sollten. Die beiden genannten Diskrepanzen in der datenorientierten Perspektive, die in den Ergebnissen der Ähnlichkeitsmaße nicht sichtbar sind, sind unterschiedlich zu bewerten. Ein unterschiedlicher Datenzugriff kann über die Ähnlichkeit in der operationalen Perspektive erkannt werden, d. h., die Diskrepanz betrifft im Grunde genommen die operationale Perspektive, nicht die datenorientierte. Es bleibt somit ein Punkt, der der Objektinstanzen, der nicht erkannt wird. Dieser kann zunächst einmal hingenommen werden, vor allem, da viele Modelliersprachen ebenfalls keine Objektinstanzen abbilden. Zu einem späteren Zeitpunkt kann über ein Ähnlichkeitsmaß auf Instanzbasis nachgedacht werden.

4.3 Abgleich der verhaltensorientierten Perspektive

Dadurch, dass bei M:N-Abbildungen mehrere Aktivitäten in einer Menge gefasst werden, die mit einer anderen Menge an Aktivitäten abgeglichen wird, und keine 1:1-Zuordnungen zwischen einzelnen Aktivitäten mehr bestehen, ist ein Abgleich des Verhaltens über Bisimulation bzw. Ausführungspfade, bei dem Aktivitäten einzeln betrachtet werden, nicht direkt übertragbar. Stattdessen wird mit dem im Folgenden zunächst beschriebenen Ansatz, der von Baumann et al. (2015a) vorgestellt und von Baumann et al. (2016c) evaluiert wird, versucht, bestimmte Informationen über das Verhalten den Aktivitäten direkt als Eigenschaften bzw. Merkmale zuzuordnen, um sie ähnlich wie Beschriftungen oder Ressourcenmengen behandeln zu können. Diese Informationen sind die relative Position einer Aktivität im Modell, die Optionalität einer Aktivität und die Wiederholbarkeit einer Aktivität. Reijers und Mendling (2011) nennen in ihrer Arbeit vier Merkmale, die das Verhalten eines Prozessmodells auszeichnen: Ausführungsreihenfolge, Ausschließlichkeit (Exklusivität), Nebenläufigkeit (Parallelität) und Wiederholbarkeit. Die ersten drei Merkmale beziehen sich hierbei immer auf die Beziehung zwischen je zwei Aktivitäten, nicht auf eine einzelne Aktivität. Die relative Position einer Aktivität (Abschnitt 4.3.1) stellt eine Annäherung an die Ausführungsreihenfolge dar, die für jede Aktivität separat angegeben werden kann, und führt auch einen neuen Aspekt bei der Analyse eines Prozessmodells ein: Wann kann eine Aktivität (zum ersten Mal) ausgeführt werden? Gegen Anfang, in der Mitte oder gegen Ende einer Prozessinstanz? Über Ausschließlichkeit oder Parallelität macht die relative Position keine Aussage. Die Optionalität von Knoten (Abschnitt 4.3.3) wird von Weidlich et al. (2010b) ebenfalls als Merkmal einer einzelnen Aktivität genannt. Die Wiederholbarkeit (Abschnitt 4.3.2) gibt Hinweise, ob eine Aktivität in einer Schleife enthalten ist. In Abschnitt 4.3.4 werden die Ähnlichkeitsmaße, die auf der Position, der Optionalität und der Wiederholbarkeit basieren, auf Varianten mit Straftermen erweitert, um inhomogene Mengenbildung bei der M:N-Abbildung zu bestrafen. Die Ähnlichkeiten von Position, Optionalität und Wiederholbarkeit können stets kombiniert werden, da sie voneinander unabhänge Aspekte der Aktivitäten erfassen und gemeinsam ein besseres Bild des Verhaltes eines Modells liefern als einzeln betrachtet.

Anschließend an diese Verhaltenseigenschaften einzelner Aktivitäten zeigt Abschnitt 4.3.5 einen weiteren Ansatz zur Ähnlichkeitsbestimmung von Prozessverhalten, der ursprünglich von Baumann et al. (2014) vorgestellt wird und eine Ordnungsrelation auf Mengen verwendet. Einige der Einschränkungen, die in der ursprünglichen Methode notwendig sind, werden in dieser Arbeit aufgehoben. Zum Schluss wird in Abschnitt 4.3.6 eine Verallgemeinerung der Abgleichsmethode mittels kausaler Verhaltensprofile (siehe Abschnitt 2.2.4.5) gezeigt, die auch auf deklarative Prozessmodelle übertragbar ist. Diese wird von Baumann (2017) vorgestellt und kann in der momentanen Form aber keine M:N-Abbildungen, sondern nur 1:1-Abbildungen berücksichtigen.

4.3.1 Positionsähnlichkeit

Da für Informationen über beispielsweise die Reihenfolge von Aktivitäten stets mehrere, d. h. mindestens zwei, Aktivitäten miteinander in Beziehung gesetzt werden müssen, wird über einen anderen Weg versucht, Informationen über den Ablauf eines Prozessmodells als Eigenschaften der jeweiligen Aktivitäten zu erfassen. Die Reihenfolge wird mit der Position eines Knotens im Prozessmodell angenähert, was eine neue, einfach zu ermittelnde Eigenschaft von Aktivitäten ist, die unabhängig von anderen Aktivitäten bestimmt werden kann.

Als Position wird die relative Position einer Aktivität auf dem kürzesten Pfad von Anfang

bis Ende, der die jeweilige Aktivität einschließt, verwendet. Das Startereignis hat stets die Position 0 und das Endereignis die Position 1. Das heißt, die Position einer Aktivität gibt an, wann eine Aktivität zum ersten Mal in einer möglichst kurzen Ausführung auftaucht. Bei einem rein sequentiellen Prozessmodell kann so eindeutig auf die Reihenfolge der Aktivitäten geschlossen werden. Sind Gateways im Modell enthalten, kann eine Reihenfolge rein aus der Position nicht einfach bestimmt werden. Eine weitere Aussage, die jedoch getroffen werden kann, ist die, dass zwei Aktivitäten mit derselben Position auf jeden Fall auf unterschiedlichen Teilpfaden liegen. Ob diese Teilpfade jedoch parallel oder exklusiv sind, kann auch hier nicht erkannt werden. Zwei Aktivitäten mit einer nahe beieinanderliegenden Position könnten sequentiell nahe beieinander liegen oder auf unterschiedlichen Teilpfaden sein. Dies zeigt, dass die Position zwar keine überaus verlässliche Angabe ist, um das Verhalten bzw. eine Reihenfolge von Aktivitäten zu beschreiben, allerdings lassen sich, wie in Abschnitt 5.1 gezeigt, empirisch dennoch gute Ergebnisse erzielen, wobei eine Verknüpfung mit Optionalität und Wiederholbarkeit angebracht ist. Des Weiteren lässt sich die Position durch das reine Abzählen von Kanten folgendermaßen einfach berechnen: Um die Position der Aktivitäten zu ermitteln, werden die Kanten gezählt, die mindestens zwischen dem Startereignis und der Aktivität liegen. Diese Anzahl wird durch die Anzahl der Kanten von Start bis Ende des kürzesten Pfades, auf dem die Aktivität liegt, geteilt. Die Position beschreibt also das relativ zur Länge des kürzesten Pfades, der von Anfang bis Ende über die jeweilige Aktivität läuft, erste Auftreten der Aktivität. Wegen erlaubter Schleifen kann keine gemittelte (relative) Position über alle die Aktivität einschließenden Pfade angegeben werden. Falls eine Aktivität jedoch mehrfach ausgeführt werden kann, so ist dies im Merkmal der Wiederholbarkeit erfasst. Die Position einer Knotenmenge, die im Weiteren als Zentroid bezeichnet wird, ist das arithmetische Mittel aus den relativen Positionen der einzelnen Knoten.

Definition 4.12 (Relative Knotenposition)**.** Gegeben ist ein Prozessmodell $G = (N, E, \lambda)$. Es bezeichne $m(n_1, n_2)$ die Länge des kürzesten Pfades von n_1 nach n_2 mit $n_1, n_2 \in N$. Die Position $\pi(n)$ eines Knotens $n \in N$ errechnet sich über

$$\pi(n) = \frac{m(e_{start}, n)}{m(e_{start}, n) + m(n, e_{end})} \in [0, 1].$$

Es gilt stets $m(n, n) = 0$ sowie $\pi(e_{start}) = 0$ und $\pi(e_{end}) = 1$. Ein Zentroid ist wie folgt definiert.

Definition 4.13 (Zentroid einer Knotenmenge)**.** Gegeben ist ein Prozessmodell $G = (N, E, \lambda)$. Es sei $P \subseteq \mathcal{P}(N)$ eine Partition gemäß Definition 3.3. Der Zentroid $\pi(p)$ einer Knotenmenge $p \in P \ni \emptyset$ ist

$$\pi(p) = \frac{1}{|p|} \sum_{n \in p} \pi(n),\ p \neq \emptyset.$$

Die leere Menge $\emptyset$ hat keine Position. Es ist $\pi(p) \in [0, 1]$.

Abbildung 4.4 zeigt ein Prozessmodell mit Positionen der einzelnen Knoten und angedeuteten Zentroiden der zu Mengen zusammengefassten Knoten. Zur Bestimmung der Position werden Gateways aus dem Modell ausgeblendet und Aktivitäten alle direkt miteinander verbunden. Dies entspricht dem Vorgehen der Methode 1 aus Abschnitt 2.2.3.4. Die Information aus den verschiedenen Arten der Gateways wird zur Bestimmung der Position nicht benötigt. Das Ausblenden der Gateways stellt dabei sicher, dass die Position nicht von beliebig

schachtelbaren Verzweigungen beeinflusst wird. So haben die Knoten/Aktivitäten der beiden Modelle aus Abbildung 4.5, trotz unterschiedlicher Gatewayschachtelung, die gleichen Positionen.

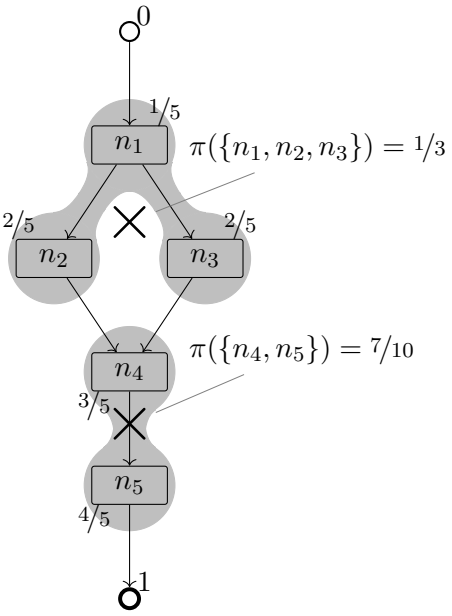


Abbildung 4.4: Positionen der einzelnen Aktivitäten und Zentroiden der Knotenmengen.

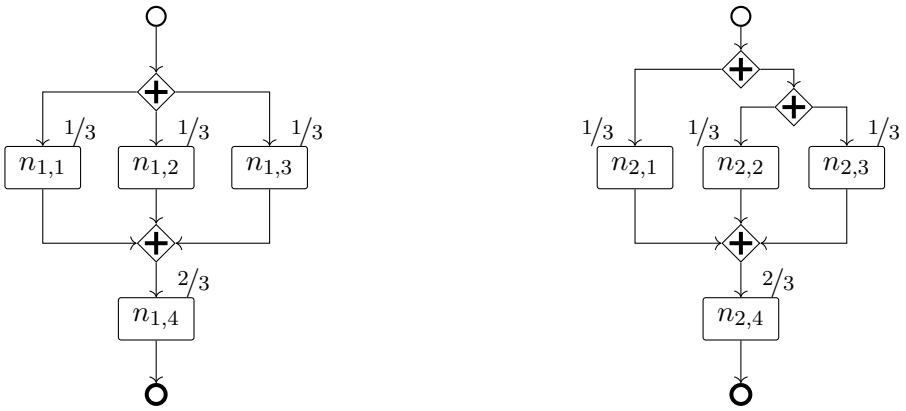


Abbildung 4.5: Trotz unterschiedlicher Gatewayschachtelung ist $\pi(n_{1,k}) = \pi(n_{2,k})$ für alle $k = 1, \ldots, 4$.

Die Ähnlichkeit zweier Prozessmodelle bezüglich ihrer Zentroiden errechnet sich über die gemittelte Differenz der Zentroiden der abgebildeten Knotenmengen. Für eine Abbildung M beschreibt $|\pi(p) - \pi(M(p))|$ die Differenz der Zentroiden von Knotenmenge p und deren Bild $M(p)$. Je größer dieser Wert, desto weiter entfernt sind die Zentroiden in den Modellen. Die Differenz nimmt Werte aus $[0, 1]$ an. Um daraus eine Ähnlichkeit zu machen, kann einfach $1 - |\pi(p) - \pi(M(p))|$ gerechnet werden. Dies ist die Zentroidähnlichkeit Sim_{cen} mit $Sim_{cen}(p_1, p_2) = 1 - |\pi(p_1) - \pi(p_2)|$. Die Zentroidähnlichkeit zweier Prozessmodelle unter einer gegebenen Abbildung M ist in Definition 4.14 angegeben.

Definition 4.14 (Zentroid-/Positionsähnlichkeit)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(A_1)$ und $P_2 \subseteq \mathcal{P}(A_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Die Zentroid- bzw. Positionsähnlichkeit von G_1 und G_2 unter M ist gegeben durch

$$\begin{aligned} VSim_M^{\pi}(G_1, G_2) &= \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} (1 - |\pi(p) - \pi(M(p))|) \\ &= \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} Sim_{cen}(p, M(p)). \end{aligned}$$

Es gilt $VSim_M^{\pi}(\cdot, \cdot) \in [0, 1]$.

In der Berechnung von $VSim_M^{\pi}(\cdot, \cdot)$ werden nur die tatsächlich abgebildeten Knotenmengen berücksichtigt, denn nur von den abgebildeten Knotenmengen kann eine Ähnlichkeit bestimmt werden. Die nicht abgebildeten Knotenmengen gehen über den Anteil der abgebildeten Knoten (siehe Definition 2.7 bzw. Definition 4.34), der dann niedriger ist, wenn mehr Knoten gelöscht werden, in die Gesamtähnlichkeitsberechnung ein.

Beispiel 4.6. Gegeben sind die beiden Prozessmodelle aus Abbildung 4.6 mit der angedeuteten Abbildung $M : P_1 \to P_2$, wobei $P_1 = \{p, q, \emptyset\}$ mit $p = \{n_{1,1}, n_{1,2}, n_{1,3}\}$ und $q = \{n_{1,4}, n_{1,5}\}$ und $P_2 = \{p', q', \{n_{2,3}\}\}$ mit $p' = \{n_{2,1}, n_{2,2}\}$ und $q' = \{n_{2,4}\}$. Es ist $M(p) = p'$, $M(q) = q'$ und $M(\emptyset) = \{n_{2,3}\}$.

Die Zentroiden der Knotenmengen sind $\pi(p) = 1/3$, $\pi(q) = 7/10$ sowie $\pi(p') = 3/8$ und $\pi(q') = 3/4$. Für die abgebildeten Paare ergibt sich eine Differenz von $|\pi(p) - \pi(p')| = 1/24 \approx 0,04$ und $|\pi(q) - \pi(q')| = 1/20 = 0,05$, was zu Ähnlichkeiten von $Sim_{cen}(p, p') = 23/24$ und $Sim_{cen}(q, q') = 19/20$ führt. Die Zentroidähnlichkeit unter dieser Abbildung ist dann

$$VSim_M^{\pi}(G_1, G_2) = \frac{1}{2}\left(\frac{23}{24} + \frac{19}{20}\right) = \frac{229}{240} \approx 0,95.$$

4.3.2 Wiederholbarkeitsähnlichkeit

Neben seiner Position kann einem Knoten auch seine Wiederholbarkeit, die mit ρ bezeichnet wird, zugewiesen werden, wobei diese nur entweder 0 oder 1 sein kann. Ein Knoten ist entweder wiederholbar oder nicht wiederholbar. Eine möglicherweise gegebene Höchstanzahl seiner Wiederholungen wird nicht mit einberechnet. Die Wiederholbarkeit als Eigenschaft von Aktivitäten wird wie auch die Position von Baumann et al. (2015a) eingeführt.

Definition 4.15 (Wiederholbarkeit eines Knotens)**.** Ein Knoten n eines Prozessmodells $G = (N, E, \lambda)$ mit $n \in N$ ist wiederholbar, wenn er mehr als einmal in einer Prozessinstanz ausgeführt werden kann: $\rho(n) = 1$. Er ist nicht wiederholbar, wenn er maximal einmal ausgeführt werden kann: $\rho(n) = 0$.

Wiederholbarkeit ist nur dann gegeben, wenn Schleifen im Modell vorhanden sind. Sind keine Schleifen im Modell, so ist kein Knoten wiederholbar. Abbildung 4.7 zeigt ein Prozessmodell mit wiederholbaren und nicht wiederholbaren Knoten.

Wie auch bei der Position kann für eine Menge an Knoten eine mittlere Wiederholbarkeit (Wiederholbarkeitszentroid) angegeben werden.

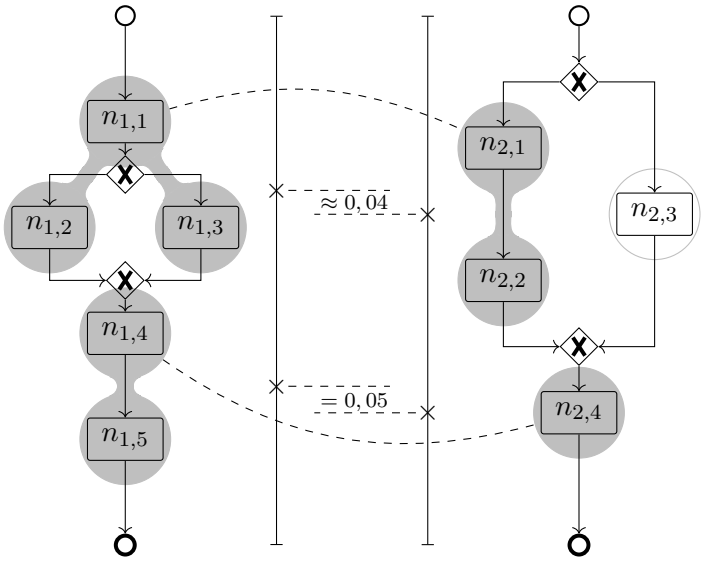


Abbildung 4.6: Illustration zur Positionsähnlichkeit zweier Prozessmodelle

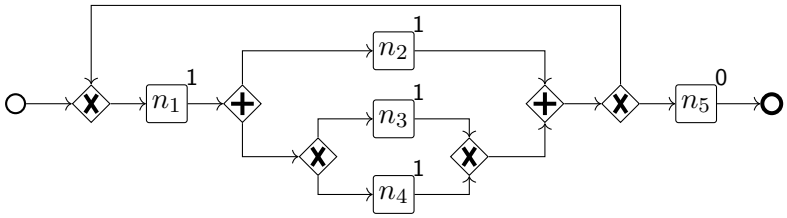


Abbildung 4.7: Prozessmodell mit angegebener Wiederholbarkeit der Aktivitäten. Wiederholbare Aktivitäten: n_1, n_2, n_3, n_4. Nur n_5 ist nicht wiederholbar.

Definition 4.16 (Wiederholbarkeitszentroid einer Knotenmenge)**.** Gegeben ist ein Prozessmodell $G = (N, E, \lambda)$. Es sei $P \subseteq \mathcal{P}(N)$ eine Partition gemäß Definition 3.3. Der Wiederholbarkeitszentroid $\rho(p)$ einer Knotenmenge $p \in P \ni \emptyset$ ist

$$\rho(p) = \frac{1}{|p|} \sum_{n \in p} \rho(n),\ p \neq \emptyset.$$

Die leere Menge $\emptyset$ hat keine Wiederholbarkeit. Es ist $\rho(p) \in [0, 1]$.

Für das Modell in Abbildung 4.7 gilt beispielsweise $\rho(\{n_1, n_2\}) = 1$, $\rho(\{n_2, n_5\}) = 0,5$ und $\rho(\{n_3, n_4, n_5\}) = 2/3$. Analog zur Positionsähnlichkeit für Knotenmengen kann auch eine Wiederholbarkeitsähnlichkeit Sim_{rep} für Knotenmengen über die Differenz der Wiederholbarkeitszentroiden berechnet werden: $Sim_{rep}(p_1, p_2) = 1 - |\rho(p_1) - \rho(p_2)| \in [0, 1]$. Eine Mittelung dieser Werte ergibt dann eine Wiederholbarkeitsähnlichkeit zweier Prozessmodelle unter einer gegebenen Abbildung.

Definition 4.17 (Wiederholbarkeitsähnlichkeit)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(A_1)$ und $P_2 \subseteq \mathcal{P}(A_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Die Wiederholbarkeitsähnlichkeit von

G_1 und G_2 unter M ist gegeben durch

$$VSim_M^{\rho}(G_1, G_2) = \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} (1 - |\rho(p) - \rho(M(p))|)$$
$$= \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} Sim_{rep}(p, M(p)).$$

Es gilt $VSim_M^{\rho}(\cdot, \cdot) \in [0, 1]$.

In den Beispielmodellen aus Abbildung 4.6 ist kein Knoten wiederholbar, d. h., die Wiederholbarkeitsähnlichkeit für diese beiden Modelle ist 1.

4.3.3 Optionalitätsähnlichkeit

Ein Knoten ist genau dann optional (siehe auch Weidlich et al., 2010b), wenn es mindestens eine Möglichkeit gibt, den Prozess auszuführen, ohne dabei den betrachteten Knoten auszuführen. Ganz konkret spielt Optionalität dann eine Rolle, sobald XOR-Gateways im Modell auftauchen. Analog zur Wiederholbarkeit ist auch die Optionalität ein boolesches Merkmal einer Aktivität.

Definition 4.18 (Optionalität eines Knotens)**.** Ein Knoten n eines Prozessmodells $G = (N, E, \lambda)$ mit $n \in N$ ist optional, wenn er in einer Prozessinstanz nicht ausgeführt werden muss: $o(n) = 1$. Er ist nicht optional (also verpflichtend), wenn er mindestens einmal ausgeführt werden muss: $o(n) = 0$. (o steht für *omikron*.)

Abbildung 4.8 zeigt ein Prozessmodell mit optionalen und nicht optionalen (verpflichtenden) Knoten. Wie auch bei der Position und der Wiederholbarkeit kann für eine Menge an Knoten eine mittlere Optionalität (Optionalitätszentroid) angegeben werden.

Definition 4.19 (Optionalitätszentroid einer Knotenmenge)**.** Gegeben ist ein Prozessmodell $G = (N, E, \lambda)$. Es sei $P \subseteq \mathcal{P}(N)$ eine Partition gemäß Definition 3.3. Der Optionalitätszentroid $o(p)$ einer Knotenmenge $p \in P \ni \emptyset$ ist

$$o(p) = \frac{1}{|p|} \sum_{n \in p} o(n), \ p \neq \emptyset.$$

Die leere Menge $\emptyset$ hat keine Optionalität. Es ist $o(p) \in [0, 1]$.

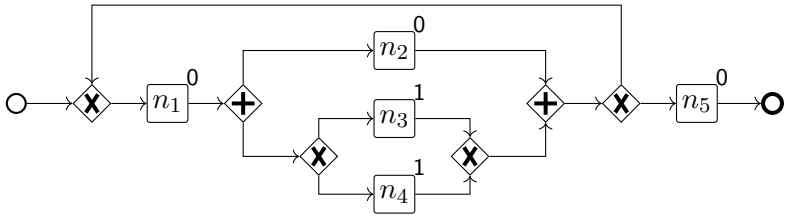


Abbildung 4.8: Prozessmodell mit angegebener Optionalität der Aktivitäten. Optionale Aktivitäten: n_3 und n_4. Aktivitäten n_1, n_2 und n_5 sind verpflichtend.

Für das Modell in Abbildung 4.7 gilt beispielsweise $o(\{n_1, n_2\}) = 0$, $o(\{n_3, n_4\}) = 1$ und $o(\{n_3, n_4, n_5\}) = 2/3$. Analog zur Positions- und Wiederholbarkeitsähnlichkeit für Knotenmengen kann auch eine Optionalitätsähnlichkeit Sim_{opt} für Knotenmengen über die Differenz der Optionalitätszentroiden berechnet werden: $Sim_{opt}(p_1, p_2) = 1 - |o(p_1) - o(p_2)| \in [0, 1]$. Eine Mittelung dieser Werte ergibt dann eine Optionalitätsähnlichkeit zweier Prozessmodelle unter einer gegebenen Abbildung.

Definition 4.20 (Optionalitätsähnlichkeit)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(A_1)$ und $P_2 \subseteq \mathcal{P}(A_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Die Optionalitätsähnlichkeit von G_1 und G_2 unter M ist gegeben durch

$$\begin{aligned} VSim^o_M(G_1, G_2) &= \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} (1 - |o(p) - o(M(p))|) \\ &= \frac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} Sim_{opt}(p, M(p)). \end{aligned}$$

Es gilt $VSim^o_M(\cdot, \cdot) \in [0, 1]$.

Beispiel 4.7. Gegeben sind die beiden Prozessmodelle aus Abbildung 4.9 mit der angedeuteten Abbildung $M : P_1 \to P_2$, wobei $P_1 = \{p, q, \emptyset\}$ mit $p = \{n_{1,1}, n_{1,2}, n_{1,3}\}$ und $q = \{n_{1,4}, n_{1,5}\}$ und $P_2 = \{p', q', \{n_{2,3}\}\}$ mit $p' = \{n_{2,1}, n_{2,2}\}$ und $q' = \{n_{2,4}\}$. Es ist $M(p) = p'$, $M(q) = q'$ und $M(\emptyset) = \{n_{2,3}\}$.

Die Optionalitätszentroiden der Knotenmengen sind $o(p) = 2/3$ und $o(q) = 0$ sowie $o(p') = 1$ und $o(q') = 0$. Für die abgebildeten Paare ergibt sich eine Differenz von $|o(p) - o(p')| = 1/3 \approx 0,33$ und $|o(q) - o(q')| = 0$, was zu Ähnlichkeiten von $Sim_{opt}(p, p') = 2/3$ und $Sim_{opt}(q, q') = 1$ führt. Die Optionalitätsähnlichkeit unter dieser Abbildung ist dann

$$VSim^o_M(G_1, G_2) = \frac{1}{2}\left(\frac{2}{3} + 1\right) = \frac{5}{6} \approx 0,83.$$

Die Optionalitätsähnlichkeit würde erhöht werden, wenn der Knoten $n_{1,1}$ aus p entfernt werden würde, denn dann wäre $Sim_{opt}(p, p') = 1 > 2/3$, was wiederum zu einer Optionalitätsähnlichkeit von $VSim^o_M(G_1, G_2) = 1$ führen würde. Der Anteil der nicht abgebildeten Knoten wäre in diesem Fall allerdings höher und somit die Ähnlichkeit insgesamt möglicherweise geringer.

Die Berechnung der Position, der Wiederholbarkeit und der Optionalität von Knoten ist im Kapitel 5 in den jeweiligen Paragrafen von Abschnitt 5.2.1.2 genauer erläutert. Im folgenden Abschnitt wird eine Erweiterung der drei Ähnlichkeiten Position, Wiederholbarkeit und Optionalität gemacht. Diese beruht auf der Bestrafung von inhomogenen Knotenmengen, wie sie von Baumann et al. (2015a) vorgeschlagen wird. Je inhomogener eine Aktivitätenmenge ist, desto größer ist der Strafterm.

4.3.4 Straffunktionen für Verhaltensmerkmale

Bei Betrachtung der Abbildung 4.10 fällt auf, dass die abgebildeten Mengen, $\{n_{1,1}, n_{1,5}\}$ und $\{n_{2,2}\}$, die gleiche relative Position haben, was sich durch die Mittelung der beiden Knotenpositionen im linken Modell ergibt. Dabei unterscheiden sich die Positionen der einzelnen

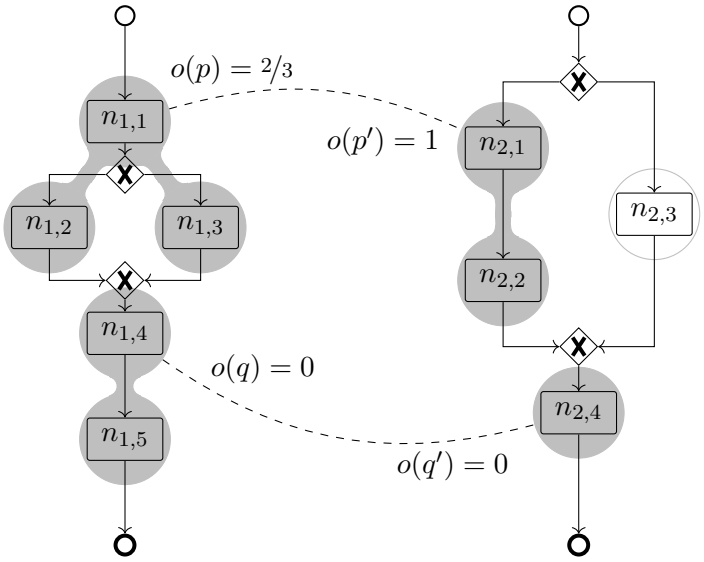


Abbildung 4.9: Illustration zur Optionalitätsähnlichkeit zweier Prozessmodelle.

Knoten doch deutlich voneinander und die Sinnhaftigkeit der Zusammenführung der ersten und letzten Aktivität im linken Modell ist, ohne zu wissen, was hinter den Aktivitäten tatsächlich steckt, fraglich. Grundsätzlich erscheint es sinnvoller, Knoten zu kombinieren, die nah beieinander liegen. Von Weidlich et al. (2010a) (siehe Abschnitt 3.3.1) wird eine Nachbarschaftseigenschaft zwingend gefordert. Mittels Einführung von Straffunktionen muss eine Nachbarschaftseigenschaft nicht unbedingt vorliegen, allerdings werden weit auseinander liegende Knotenmengen, wie in Abbildung 4.10 im linken Modell, bestraft. Genauer: Ihre Ähnlichkeit zu egal welcher anderen Knotenmenge wird um einen bestimmten Wert gesenkt, der abhängt vom Grad der Inhomogenität der Knotenmenge. Diese Art der Straffunktion wird von Baumann et al. (2015a) neben der Position als Annäherung der Reihenfolge sowie der Wiederholbarkeit und der Optionalität als weitere Verhaltensmerkmale einzelner Aktivitäten erstmals eingeführt.

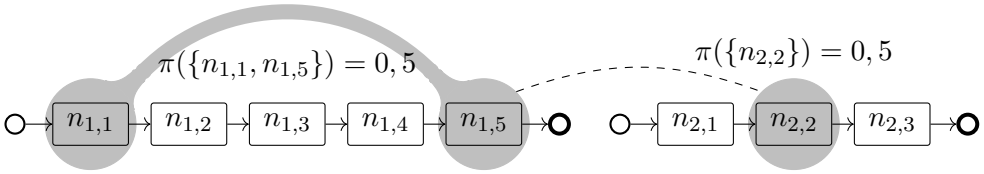


Abbildung 4.10: Illustration zur Inhomogenität von Knotenmengen.

Inhomogenität bezogen auf die Position bedeutet, dass die Knoten im Prozessmodell weit auseinander liegen. Größtmögliche Inhomogenität bezogen auf Wiederholbarkeit bzw. Optionalität liegt dann vor, wenn genau ein wiederholbarer bzw. optionaler Knoten in der Knotenmenge liegt und ein nicht wiederholbarer bzw. nicht optionaler, die Wiederholbarkeits-/Optionalitätszentroiden also bei genau zwei Elementen in der Menge einen Wert von 0,5 aufweisen. Inhomogenität von Mengen kann beispielsweise mittels der Streuung der Elemente angegeben werden, wobei als Maß der Streuung im Folgenden die Varianz (eigentlich die (korrigierte) Stichprobenvarianz als Schätzer der Varianz) verwendet wird. Für jede Knotenmenge

muss die Varianz für alle drei Verhaltensmerkmale berechnet werden:

Definition 4.21 (Varianz der Verhaltensmerkmale (Straffunktion))**.** Es sei $\xi \in \{\pi, \rho, o\}$. Für ein Prozessmodell $G = (N, E, \lambda)$ mit Partition P errechnet sich für $p \in P$ mit $|p| \geq 2$ die erwartungstreue Varianz bezüglich eines Merkmals ξ mittels

$$pen^{\xi}(p) = \frac{1}{|p| - 1} \sum_{n \in p} (\xi(n) - \xi(p))^2 .$$

Für p mit $|p| = 1$ ist die Angabe einer Straffunktion nicht sinnvoll, da einelementige Mengen nicht inhomogen sein können. Hier wird $pen^{\xi}(p) = 0$ gesetzt.

Die bestraften Zentroidähnlichkeiten sind damit

Definition 4.22 (Bestrafte Verhaltensähnlichkeiten)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(N_1)$ und $P_2 \subseteq \mathcal{P}(N_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Sei außerdem $\xi \in \{\pi, \rho, o\}$. Die bestraften Ähnlichkeiten von G_1 und G_2 bezüglich ξ unter M sind gegeben durch

$$\begin{aligned} &penVSim^{\xi}_{M}(G_1, G_2) \\ &= \tfrac{1}{|\{p \in P_1 | p \neq \emptyset \neq M(p)\}|} \sum_{\substack{p \in P_1 \setminus \emptyset: \\ M(p) \neq \emptyset}} \left(\underbrace{(1 - |\xi(p) - \xi(M(p))|)}_{\text{Ähnlichkeit}} - \underbrace{\left(pen^{\xi}(p) + pen^{\xi}(M(p))\right)}_{\text{Summe der Strafterme}} \right)^{+} \end{aligned}$$

Die Funktion $(\cdot)^{+}$ ist eine Kurzschreibweise für $\max\{0, \cdot\}$. Sie wird hier benötigt, da durch die Differenz nicht garantiert werden kann, dass alle Summanden ≥ 0 sind. Dies ist aber wünschenswert, da die Ähnlichkeit der einzelnen Knotenmengen nicht negativ sein soll und die Summe über alle solchen Ähnlichkeiten, nach der Definition eines Ähnlichkeitsmaßes, nicht negativ sein darf. Es lässt sich leicht nachrechnen, dass $pen^{\xi}(\cdot)$ immer einen Wert im Intervall $[0, 1/2]$ annimmt für alle ξ.

Beispiel 4.8. Der Strafterm des Positionsmerkmals für die eingezeichnete Knotenmenge $p = \{n_{1,1}, n_{1,5}\}$ im linken Modell in Abbildung 4.10 beläuft sich auf

$$\begin{aligned} pen^{\pi}(p) &= \frac{1}{2-1} \left((\pi(\{n_{1,1}\}) - \pi(p))^2 + (\pi(\{n_{1,5}\}) - \pi(p))^2 \right) \\ &= \left(\frac{1}{6} - \frac{1}{2} \right)^2 + \left(\frac{5}{6} - \frac{1}{2} \right)^2 \\ &= \frac{1}{9} + \frac{1}{9} \\ &= \frac{2}{9} \\ &\approx 0,22 \end{aligned}$$

Die bestrafte Ähnlichkeit des abgebildeten Mengenpaares berechnet sich damit zu

$$\begin{aligned} penSim_{cen}(p, M(p)) &= (Sim_{cen}(p, M(p)) - pen^{\pi}(p))^{+} \\ &= 1 - \frac{2}{9} \\ &\approx 0,78 \end{aligned}$$

Der Ähnlichkeitswert in Beispiel 4.8 erscheint, absolut gesehen, immer noch recht hoch. Dies liegt zum einen daran, dass eine Straffunktion maximal den Wert 0,5 annehmen kann, zum anderen gilt für die Positionsähnlichkeit, dass sie per Definition nicht das komplette Intervall $[0, 1]$ abdeckt. Dadurch, dass Anfangs- und Endknoten die Positionen 0 und 1 belegen, ist der Abstand zweier Aktivitäten immer < 1, also die Ähnlichkeit immer > 0, aber niemals $= 0$. Bei wenigen Knoten im Modell bzw. bei kurzen Pfaden wirkt sich dieser Effekt stärker auf den Ähnlichkeitswert aus als bei längeren Modellen. Dies ist ein ähnliches Problem wie mit der Transformation eines Distanzmaßes in ein Ähnlichhkeitsmaß gemäß Gleichung (2.2), das nie den Wert 0 annehmen kann. Überdies kann die Varianz grundsätzlich größere Werte annehmen, wenn weniger Knoten in einer Menge sind. Da beim Umgang mit Ähnlichhkeiten jedoch selten absolute Ähnlichkeitswerte benötigt werden, sondern im Normalfall Ähnlichkeitswerte miteinander verglichen werden, unter anderem bei der Suche nach dem größten Ähnlichkeitswert, wirkt sich diese Eigenschaft der Positionsähnlichkeit nicht auf das Ergebnis aus. Sind doch absolute Ähnlichkeitswerte notwendig, so ist über eine Reskalierung des Positionsähnlichkeitsmaßes nachzudenken. Für die Wiederholbarkeits- und Optionalitätsähnlichkeit gilt dies übrigens nicht. Hier können die Werte das komplette Intervall $[0, 1]$ abdecken.

Das Ähnlichkeitsmaße für das Prozessverhalten, die im Folgenden noch vorgestellt werden, setzen immer mindestens zwei Aktivitäten bzw. Mengen an Aktivitäten innerhalb eines Modells in Beziehung. Dies macht sie aufwändiger in der Berechnung als die bis dahin vorgestellten Maße, die das Verhalten auf Merkmale einzelner Aktivitäten abstrahieren.

4.3.5 Ähnlichkeit mittels Ordnungsrelationen auf Mengen

Der Ansatz, der in diesem Abschnitt vorgestellt wird, betrachtet die Reihenfolge von Knoten in den gegebenen Prozessmodellen. Im ursprünglich vorgeschlagenen Ansatz von Baumann et al. (2014) werden nur Prozessmodelle ohne Verzweigungen zugelassen, d. h. nur sequentielle Prozessmodelle. Diese Einschränkung muss im Folgenden dadurch, dass die Abgleichsmethode an einigen Stellen erweitert wird, nicht gefordert werden. Schleifen in Prozessmodellen sind grundsätzlich erlaubt, allerdings ist die Aussagekraft der Messmethode, wie bei der Definition der Ordnungsrelationen ersichtlich sein wird, nicht mehr sehr hoch.

Das Ähnlichkeitsmaß nutzt eine Ordnungsrelation von Mengen und die Erhaltung oder Nichterhaltung dieser Ordnung auf den abgebildeten Mengen aus. Hierzu wird zunächst eine Reihenfolge von Knoten mittels der gegebenen Kanten definiert: Sei $G = (N, E, \lambda)$ ein Prozessmodell mit $n', n'' \in N$. Knoten n'' folgt auf n', $n' \to n''$, falls $\exists n_1, \ldots, n_k \in N$ mit $\{(n', n_1), (n_1, n_2), \ldots, (n_{k-1}, n_k), (n_k, n'')\} \in E$. Falls $n' \to n''$ und $n'' \to n'$, dann schreibe $n' \leftrightarrow n''$. Falls weder $n' \to n''$ noch $n'' \to n'$ gilt, so schreibe $n' \nleftrightarrow n''$, d. h., n' und n'' stehen in keiner Reihenfolge. Dann ist eine Ordnungsrelation auf P wie folgt gegeben, wobei P eine Partition von N ist:

Definition 4.23 (Ordnungsrelationen auf Mengen)**.** Es sei $G = (N, E, \lambda)$ ein Prozessmodell mit Partition $P \subseteq \mathcal{P}(N)$. Für $p, q \in P$, $p, q \neq \emptyset$, gelte die schwache Ordnung

$$p \prec q \quad \Leftrightarrow \quad \forall n \in p, m \in q : \neg(m \to n) \wedge \exists n' \in p, m' \in q : n' \to m'.$$

Diese schwache Ordnungsrelation wird wie folgt zu einer starken Ordnungsrelation verschärft:

$$p \prec\prec q \quad \Leftrightarrow \quad \forall n \in p, m \in q : n \to m \wedge \nexists n' \in p, m' \in q : m' \to n'$$

Zudem können zwei Mengen auch in keiner Ordnung stehen:

$$p \sim q \quad \Leftrightarrow \quad \forall n \in p, m \in q : n \nleftrightarrow m \\ \vee \exists n \in p, m \in q : n \to m \wedge \exists n' \in p, m' \in q : m' \to n'$$

Es gilt, dass $p \prec\!\prec q \Rightarrow p \prec q$. Außerdem gelten folgende Transitivitäten:

$$p \prec\!\prec q \wedge q \prec\!\prec r \Rightarrow p \prec\!\prec r$$
$$p \prec q \wedge q \prec\!\prec r \Rightarrow p \prec r$$
$$p \prec\!\prec q \wedge q \prec r \Rightarrow p \prec r$$

Außerdem gilt im Allgemeinen:

$$p \prec q \wedge q \prec r \not\Rightarrow p \prec r$$

Die Transitivitäten können leicht mit Widerspruchsbeweisen gezeigt werden, ebenso findet sich schnell ein Gegenbeispiel für die nichtgeltende Folgerung. Die angegebenen Schlüsse entsprechen der üblichen Verwendung von „schwachen“ und „starken“ Beziehungen.

Beispiel 4.9. Abbildung 4.11 zeigt ein Beispielmodell mit Knotenmengen, für die verschiedene Ordnungsrelationen gelten.

- Für $p = \{n_1, n_3\}$ und $q = \{n_2\}$ gilt: $p \prec q$, da $n_1 \to n_2$, $\neg(n_2 \to n_1)$ und $n_3 \nleftrightarrow n_2$
- Für $p = \{n_1, n_2\}$ und $q = \{n_5\}$ gilt: $p \prec\!\prec q$, da $n_1 \to n_5$, $n_2 \to n_5$ und $\neg(n_5 \to n_1)$, $\neg(n_5 \to n_2)$
- Für $p = \{n_3\}$ und $q = \{n_4\}$ gilt: $p \sim q$, da weder $n_3 \to n_4$ noch $n_4 \to n_3$

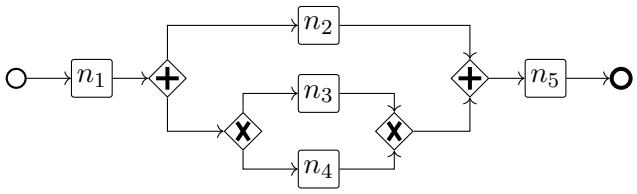


Abbildung 4.11: Prozessmodell zur Illustration verschiedener Ordnungsrelationen.

Für die definierten Ordnungsrelationen kann nun festgestellt werden, ob diese unter der Abbildung M bestehen bleiben, abgeschwächt werden, aufgehoben werden oder auf irgendeine Weise vertauscht werden. Für jede Situation wird ein Wert zwischen 0 und 1 zugewiesen. Diese Werte sind keine Ähnlickeitswerte im eigentlichen Sinn, aus ihnen wird aber durch Mittelung ein Ähnlichkeitswert berechnet. Folglich ist eine Ordnungserhaltung unter M mit hohen Werten und eine Ordnungsumkehr mit Werten bei 0 einzustufen. Eine allgemeine Belegung dieser Werte ist in Tabelle 4.2 gegeben, wobei für die Parameter $1 = a \geq b \geq c \geq d \geq e \geq f = 0$ gilt.

Die Umkehr einer starken Ordnung soll auf jeden Fall mit dem schlechtmöglichsten Wert belegt werden und die Beibehaltung einer starken Ordnung mit dem bestmöglichen Wert. Dazwischen können Abstufungen getroffen werden. Eine mögliche Parameterbelegung ist $a =$

Tabelle 4.2: Allgemeine Bewertungstabelle für den Abgleich von Ordnungsrelationen.

$Sim_{ord,M}(p,q)$	$p \prec\prec q$	$p \prec q$	$p \sim q$	$p \succ q$	$p \succ\succ q$
$M(p) \prec\prec M(q)$	a	b	d	e	f
$M(p) \prec M(q)$	b	b	c	e	e
$M(p) \sim M(q)$	d	c	c	c	d
$M(p) \succ M(q)$	e	e	c	b	b
$M(p) \succ\succ M(q)$	f	e	d	b	a

$1, b = 0,8, c = 0,6, d = 0,4, e = 0,2, f = 0$. Eine strengere Belegung wäre $a = 1, b = 0,75, c = 0,5, d = 0,25, e = f = 0$.

Aus diesen Bewertungen der Ordnungserhaltung unter der gegebenen Abbildung wird die Ähnlichkeit der abgebildeten Aktivitätenmengen über eine Mittelung der Werte aller möglichen Mengenpaare und deren abgebildeter Paare bestimmt:

Definition 4.24 (Verhaltensähnlichkeit mittels Ordnungserhaltung)**.** Es seien $G_1 = (N_1, E_1, \lambda_1)$ und $G_2 = (N_2, E_2, \lambda_2)$ zwei Prozessmodelle mit Partitionen $P_1 \subseteq \mathcal{P}(N_1)$ und $P_2 \subseteq \mathcal{P}(N_2)$. Es sei $M : P_1 \to P_2$ eine Abbildung gemäß Definition 3.3. Für alle Elemente $p, q \in P_1$ sei mittels $Sim_{ord,M}(p,q)$ eine Bewertung der Ordnungserhaltung gemäß Definition 4.23 und Tabelle 4.2 gegeben. Die Verhaltensähnlichkeit mittels der Ordnungserhaltung ist gegeben durch

$$VSim_M^{ord}(G_1, G_2) = \frac{1}{|\{p \neq q \in P_1 : p \neq \emptyset \neq q, M(p) \neq \emptyset \neq M(q)\}|} \sum_{\substack{p \neq q \in P_1: \\ p \neq \emptyset \neq q, \\ M(p) \neq \emptyset \neq M(q)}} Sim_{ord,M}(p,q).$$

Es gilt, dass $VSim_M^{ord}(G_1, G_2) \in [0,1]$.

Sobald Schleifen im Prozessmodell auftauchen, ist zu erwarten, dass viele, womöglich sogar alle, Aktivitätenmengen in Relation $\sim$ zueinander stehen. Das Ähnlichkeitsmaß besitzt dann keine Aussagekraft mehr. Es kann sinnvollerweise nur für Modelle ohne oder mit wenigen „kleinen“ Schleifen eingesetzt werden.

4.3.6 Ähnlichkeitsabgleich über Flussabhängigkeiten

Die Abgleichsmethode, die in diesem Abschnitt gezeigt wird, verwendet Flussabhängigkeiten (Jablonski, 2010) in Prozessmodellen, d. h. nicht nur den Kontrollfluss, sondern auch Abhängigkeiten, die sich beispielsweise über den Datenfluss ergeben. Die Zuordnung von Datenobjekten zu Aktivitäten und der Fluss der Datenobjekte sind oftmals eine Begründung für den Kontrollfluss zwischen Aktivitäten (Liu et al., 2007), der sich aus dem Datenfluss und evtl. weiterer Vorgaben ergibt. Im Folgenden werden lokale Abhängigkeiten, das sind Abhängigkeiten zwischen je zwei Aktivitäten, betrachtet. Diese werden aus dem Kontrollfluss, aber auch aus dem Datenfluss extrahiert. Xing et al. (2013), die Abhängigkeiten in BPEL-Modellen untersuchen, verwenden ebenfalls diese Unterscheidung. Die kausalen Verhaltensprofile, wie sie von Weidlich et al. (2010b) definiert wurden (siehe Abschnitt 2.2.4.5), basieren dagegen rein auf dem Kontrollfluss. Für eine Ähnlichkeitsberechnung sind in diesem Abschnitt, da Abhängigkeiten nur zwischen einzelnen Aktivitäten bestimmt werden, nur 1:1-Abbildungen zugelassen.

In Abschnitt 4.3.6.1 werden zunächst ordnende und kausale Abhängigkeiten zwischen Aktivitäten definiert. Ordnende Abhängigkeiten beziehen sich auf die vorgegebene oder mögliche Reihenfolge der Aktivitäten, während kausale Abhängigkeiten sich auf Existenzbeziehungen berufen. Anschließend wird in Abschnitt 4.3.6.2 gezeigt, wie aus imperativen Prozessmodellen die beiden Arten an Abhängigkeiten aus dem Kontrollfluss abgeleitet werden können. In Abschnitt 4.3.6.3 wird eine Verfeinerung auf Basis von Datenflussinformationen vorgenommen. Als nächstes wird in Abschnitt 4.3.6.4 gezeigt, wie die abgeleiteten Abhängigkeiten mittels teilweiser Hierarchiebildung miteinander verglichen werden. In diesem Schritt können bereits Aussagen über Enthaltenseinsbeziehungen zwischen Modellen gemacht werden. Zum Schluss wird in Abschnitt 4.3.6.5 ein Ähnlichkeitsmaß auf Basis der Hierarchiebildung im vorigen Schritt hergeleitet.

4.3.6.1 Ordnende und kausale Abhängigkeiten zwischen Aktivitäten

Im Folgenden werden die verschiedenen Möglichkeiten an Abhängigkeiten eingeführt. Dabei wird zwischen zwei grundsätzlichen Typen an Abhängigkeiten unterschieden: ordnende Abhängigkeiten, die rein auf einer Reihenfolge von Aktivtäten basieren, und kausale Abhängigkeiten, die vor allem Existenzbedingungen zwischen Aktivitäten ausdrücken. Zwischen je zwei Aktivitäten kann sowohl eine ordnende als auch eine kausale Abhängigkeit, auch Existenzabhängigkeit genannt, bestimmt werden. Von Jablonski (1994) wird ebenfalls eine solche Unterscheidung gemacht und Dourish et al. (1996) sprechen in ihrer Arbeit von *dependencies* für die kausalen Zusammenhänge und *temporal sequence* für die Reihenfolge der Aktivitäten. Die ordnenden Abhängigkeiten machen keine Aussage darüber, ob die Ordnung direkt erfolgen muss, also wann genau eine Aktivität auf eine andere folgen muss.

Definition 4.25 (Abhängigkeitstypen)**.** Zwischen je zwei voneinander verschiedenen Aktivitäten eines Prozessmodells gibt es folgende Abhängigkeiten, wobei jedem Aktivitätenpaar sowohl eine ordnende als auch eine kausale Abhängigkeit zugeordnet werden kann.

	Abhängigkeit	Symbol
ordnend	verpflichtende Ordnung	$\twoheadrightarrow$
	optionale Ordnung	$\rightarrow$
	keine (erlaubte) Ordnung	$-$
kausal	symmetrische Existenzabhängigkeit	$==$
	asymmetrische Existenzabhängigkeit	$=>$
	exklusive Existenzabhängigkeit	$><$
	keine (spezifizierte) Existenzabhängigkeit	$\sim$

Die angegebenen Symbole sind für zwei Aktivitäten a und b, $a \neq b$, folgendermaßen zu lesen:

- $\twoheadrightarrow (a, b)$: Nach Aktivität a muss immer (irgendwann) Aktivität b ausgeführt werden.
- $\rightarrow (a, b)$: Nach Aktivität a kann (irgendwann) Aktivität b ausgeführt werden.
- $\twoheadleftarrow (a, b)$: Vor Aktivität b muss immer (irgendwann) Aktivität a ausgeführt werden bzw. worden sein.
- $\leftarrow (a, b)$: Vor Aktivität b kann (irgendwann) Aktivität a ausgeführt werden bzw. worden sein.

- $-(a, b)$: Vor Aktivität b darf nicht Aktivität a gemacht werden und nach Aktivität a nicht Aktivität b (a und b können in keiner Instanz in dieser Reihenfolge hintereinander ausgeführt werden).

Durch eine Kombination der ordnenden Abhängigkeiten, die für jedes Aktivitätenpaar eine Vor- und eine Rückschaubedingung angeben oder keine Ordnung erlauben, lassen sich insgesamt fünf verschiedene zusammengesetzte ordnende Abhängigkeiten bilden. Diese relativ detaillierte Unterscheidung der ordnenden Abhängigkeiten erfolgt vor allem in Hinblick auf die Übertragung auf deklarative Prozessmodelle (Abschnitt 4.5.3.4) und erlaubt auf einfache Weise eine differenzierte Betrachtung unterschiedlicher binärer Abhängigkeiten:

- $\twoheadleftarrow\!\!\rightarrow (a, b)$: Nach a muss b gemacht werden und vor b muss a gemacht werden.
- $\twoheadleftarrow (a, b)$: Vor b muss a ausgeführt werden, aber nach a muss nicht zwingend b folgen.
- $\leftarrow\!\!\twoheadrightarrow (a, b)$: Nach a muss zwingend b gemacht werden, aber um b zu machen, muss nicht unbedingt a vorher ausgeführt werden.
- $\leftrightarrow (a, b)$: Nach a kann b gemacht werden und vor b kann a gemacht werden, es besteht aber keine Verpflichtung.
- $-(a, b)$: Diese Abhängigkeit ist gleichzeitig schon Vor- und Rückschaubedingung und wie oben zu lesen.

Auch für (b, a) kann eine der fünf genannten ordnenden Abhängigkeiten bestimmt werden. Es kann z. B. $-(a, b)$ und $\twoheadleftarrow\!\!\rightarrow (b, a)$ gelten. Die Symbole der kausalen Abhängigkeiten sind folgendermaßen zu lesen:

- $=> (a, b)$: Wenn Aktivität a gemacht wird, muss auch Aktivität b (irgendwann) gemacht werden.
- $== (a, b)$: Wenn Aktivität a gemacht wird, muss auch Aktivität b (irgendwann) gemacht werden und umgekert. $== (a, b)$ ist die Kombination aus $=> (a, b)$ und $=> (b, a)$.
- $>< (a, b)$: Wenn Aktivität a gemacht wird, darf in derselben Instanz nicht auch Aktivität b gemacht werden und umgekehrt.
- $\sim (a, b)$: Zwischen a und b ist keine der obigen drei kausalen Abhängigkeiten spezifiziert.

Im Unterschied zu den ordnenden Abhängigkeiten ist bei den kausalen keine Reihenfolge spezifiziert. Das heißt, wenn die kausale Abhängigkeit für (a, b) bestimmt ist, ist automatisch auch die für (b, a) bekannt, weil es keine Vor- und Nachbedingungen gibt, was bei den ordnenden Abhängigkeiten nicht der Fall ist, denn dort gibt es die Unterscheidung in Vor- und Nachbedingung. Wenn beispielsweise $=> (a, b)$ gilt, dann gilt auch automatisch $<= (b, a)$, nicht aber zwingend $=> (b, a)$. Die Abhängigkeiten $==$, $><$ und $\sim$ sind symmetrisch.

Die ordnenden und kausalen Abhängigkeiten sind nicht komplett unabhängig voneinander, denn aus der verpflichtenden Ordnung $\twoheadrightarrow (a, b)$ folgt immer die asymmetrische Existenzabhängigkeit $=> (a, b)$ und aus der exklusiven Existenzabhängigkeit $>< (a, b)$ folgt, dass für a und b keine Reihenfolge angegeben werden kann: $-(a, b)$ und $-(b, a)$. Alle anderen ordnenden und kausalen Abhängigkeiten können in jeder Kombination vorliegen.

Anmerkung Die oben definierten Flussabhängigkeiten stellen eine echte Verfeinerung der von Weidlich et al. (2010b) gezeigten Relationen zur Bestimmung des kausalen Verhaltensprofils dar. Die Relationen aus Abschnitt 2.2.4.5 und deren Entsprechungen in ordnenden Flussabhängigkeiten sind in Tabelle 4.3 dargestellt.

Tabelle 4.3: Relationen des kausalen Verhaltensprofils (BP) und deren mögliche Entsprechungen in ordnenden Flussabhängigkeiten (FD).

	$\leftrightarrow (a,b)$	$\twoheadleftarrow\!\!\rightarrow (a,b)$	$\leftarrow\!\!\twoheadrightarrow (a,b)$	$\twoheadleftarrow\!\!\twoheadrightarrow (a,b)$	$-(a,b)$
$\leftrightarrow (b,a)$	$\|\|(a,b)$	$\|\|(a,b)$	$\|\|(a,b)$	$\|\|(a,b)$	$\rightsquigarrow (b,a)$
$\twoheadleftarrow\!\!\rightarrow (b,a)$	$\|\|(a,b)$	$\|\|(a,b)$	$\|\|(a,b)$	$\|\|(a,b)$	$\rightsquigarrow (b,a)$
$\leftarrow\!\!\twoheadrightarrow (b,a)$	$\|\|(a,b)$	$\|\|(a,b)$	$\|\|(a,b)$	$\|\|(a,b)$	$\rightsquigarrow (b,a)$
$\twoheadleftarrow\!\!\twoheadrightarrow (b,a)$	$\|\|(a,b)$	$\|\|(a,b)$	$\|\|(a,b)$	$\|\|(a,b)$	$\rightsquigarrow (b,a)$
$-(b,a)$	$\rightsquigarrow (a,b)$	$\rightsquigarrow (a,b)$	$\rightsquigarrow (a,b)$	$\rightsquigarrow (a,b)$	$+(a,b)$

Kausale Flussabhängigkeiten lassen sich, außer im Fall $+(a,b)$ nicht eindeutig herleiten. Die Ausschließlichkeitsrelation $+(a,b)$ impliziert die kausale Flussabhängigkeit $>\!<\ (a,b)$. Außerdem folgt aus der ordnenden Flussabhängigkeit $\twoheadrightarrow (a,b)$ die kausale Abhängigkeit $=\!>$ (a,b). Für die in Abschnitt 2.2.4.5 ebenfalls beschriebene Kookkurrenzrelation $\gg (a,b)$ ist $=\!> (a,b)$ die direkte Entsprechung. Die kausale Abhängigkeit als Kombination von $\rightsquigarrow (a,b)$ und $\gg (a,b)$ ist in Flussabhängigkeiten ausgedrückt dann $=\!> (a,b)$, $\leftarrow\!\!\twoheadrightarrow (a,b)$ und $-(b,a)$ oder $=\!> (a,b)$, $\twoheadleftarrow\!\!\twoheadrightarrow (a,b)$ und $-(b,a)$, d. h., dass insbesondere Vorgängerbedingungen über kausale Verhaltensprofile nicht erfassbar sind.

Auch die in Abschnitt 2.2.4.6 gezeigten Verhaltensprofile (Erweiterung mit umgekehrter Ordnungsrelation und verschränkter Ordnungsrelation) finden sich in den Flussabhängigkeiten wieder. Diese von Kunze et al. (2011) vorgeschlagenen Relationen berücksichtigen jedoch keine Kausalität, sondern leiten sich rein aus der Ordnung der Aktivitäten ab.

4.3.6.2 Ableitung von Abhängigkeiten aus imperativen Prozessmodellen

Es wird zunächst ein einzelnes Prozessmodell betrachtet. Um die Abhängigkeiten für jedes Paar an Aktivitäten bestimmen zu können, wird das Prozessmodell, wie von Polyvyanyy et al. (2009) vorgeschlagen, bezüglich seiner Struktur untersucht. Hierfür wird ein Prozessmodell in *Fragmente* zerlegt, die mit Hilfe des Konzepts der *Dominanz* bestimmt werden.

Definition 4.26 (Dominanz)**.** Gegeben sei ein Prozessmodell $G = (N, E, \lambda)$. Ein Knoten $n_1 \in N$ *dominiert* einen anderen Knoten $n_2 \in N$, wenn alle Pfade von e_{start} zu n_2 über n_1 gehen. Ein Knoten n_3 *postdominiert* einen anderen Knoten n_4, wenn alle Pfade von n_4 zu e_{end} über n_3 gehen.

Definition 4.27 (Prozessfragment)**.** Ein Prozessfragment eines Prozessmodells $G = (N, E, \lambda)$ ist ein Subprozess von G, der durch ein Split- und ein Joingateway g_s bzw. g_j, $g_s \in \{XOR_s, AND_s\}, g_j \in \{XOR_j, AND_j\}$, desselben Typs bestimmt wird, wobei Gateway g_s Gateway g_j dominiert und Gateways g_j Gateway g_s postdominiert, oder, im Falle von Schleifen, g_j g_s dominiert und g_s g_j postdominiert und g_j von g_s aus erreichbar ist. In allen Fällen enthält jede Schleife entweder sowohl g_s als auch g_j oder keinen der beiden Knoten. Ein Knoten n gehört zum Prozessfragment, das durch (g_s, g_j) bestimmt wird, wenn g_s n dominiert und g_j n postdominiert; zusätzlich gehört ein Knoten m zu einem Loop-Fragment, wenn g_j m dominiert und g_s m postdominiert. Prozessfragmente können geschachtelt sein, z. B. kann Fragment f_1

ein Kind von Fragment f_2 sein. Fragmente können AND- oder XOR-Fragmente sein, je nach Typ der sie bestimmenden Gateways. Ein besonderes, zusätzliches Fragment ist das komplette Prozessmodell (triviales Fragment), das durch e_{start} und e_{end} bestimmt ist. Jedem Knoten des Prozessmodells kann eine *Tiefe*, die auf der Schachtelungstiefe der Fragmente basiert, zugewiesen werden. Knoten, die nur zum trivialen Fragment gehören, haben Tiefe 0.

Beispiel 4.10. Betrachte das Beispielmodell in Abbildung 4.12. Das Modell enthält das triviale Fragment (e_{start}, e_{end}), in dem alle Aktivitäten A bis H und alle Gateways liegen. Aktivität A liegt in keinem weiteren Fragment und hat Tiefe 0. Ein weiteres Fragment ist das XOR-Fragment, in dem nur die Aktivitäten B und C liegen. Diese beiden Aktivitäten liegen in keinem weiteren Fragment und haben deswegen jeweils Tiefe 1. Im nächsten XOR-Fragment liegen die Aktivitäten D, E, F, G und H, wobei nur D, E und F in keinem weiteren Fragment liegen und deswegen ebenfalls Tiefe 1 haben. Die Aktivitäten G und H liegen in einem AND-Fragment, das im zweiten XOR-Fragment enthalten ist, also ein Kind von diesem ist, und haben deswegen Tiefe 2.

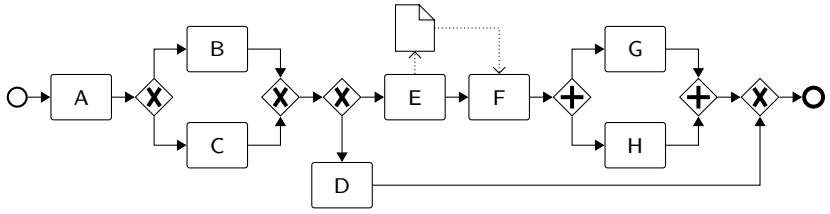


Abbildung 4.12: Beispiel eines imperativen Prozessmodells zur Bestimmung der Fragmente.

Bevor einem Aktivitätenpaar eine Abhängigkeit zugewiesen werden kann, müssen noch zwei Begriffe definiert werden:

Definition 4.28 (Parallelität zweier Aktivitäten)**.** Zwei Aktivitäten sind *optional parallel*, wenn sie zum selben AND-Fragment, aber zu unterschiedlichen Zweigen gehören, d. h., wenn innerhalb des AND-Fragments kein Pfad zwischen den beiden Aktivitäten besteht. Zwei Aktivitäten sind *verpflichtend parallel*, wenn sie optional parallel sind und beide nicht auf Pfaden von einem XOR-Split g_s zu einem korrespondierenden XOR-Join g_j innerhalb des AND-Fragments liegen.

Die letzte Bedingung in Definition 4.28 schließt diejeningen Aktivitäten nicht aus der Eigenschaft *verpflichtend parallel* aus, die auf dem (nicht optionalen) Pfad zwischen Join-Gateway und Split-Gateway eines XOR-Loop liegen. Jedes AND-Fragment hat immer mindestens zwei parallele Zweige und jedes XOR-Fragment hat immer mindestens zwei exklusive Zweige. In Abbildung 4.13 sind beispielsweise die folgenden (Nicht-)Parallelitäten zu finden: A und B sowie A und C sind nicht parallel, A und E sind verpflichtend parallel, C und E sind optional parallel. Mit diesen Begriffen können nun die ordnenden Abhängigkeiten zwischen je zwei Aktivitäten eines Prozessmodells definiert werden.

Definition 4.29 (Ordnende Abhängigkeiten im imperativen Modell)**.** Es sei $G = (N, E, \lambda)$ ein Prozessmodell und a und b zwei Aktivitäten dieses Modells. Für das Tupel (a, b) gilt

- $\twoheadrightarrow (a, b)$, falls b Aktivität a postdominiert.
- $\rightarrow (a, b)$, falls es einen Pfad von a nach b gibt, aber b a nicht postdominiert.

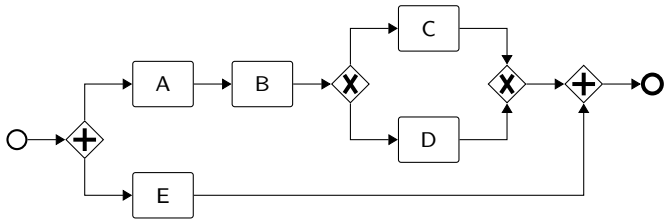


Abbildung 4.13: Beispiel eines imperativen Prozessmodells zur Veranschaulichung der verschiedenen Arten der Parallelität.

- $\twoheadleftarrow (a, b)$, falls a Aktivität b dominiert.
- $\leftarrow (a, b)$, falls es einen Pfad von a nach b gibt, aber a b nicht dominiert.
- $\leftrightarrow (a, b)$, falls a und b (optional oder verpflichtend) parallel sind. Auch aus den einzelnen Zuweisungen $\rightarrow (a, b)$ und $\leftarrow (a, b)$ wird $\leftrightarrow (a, b)$ hergeleitet.
- $-(a, b)$, wenn keine optionale oder verpflichtende Ordnung für a und b festgestellt werden kann.

Die kausalen Abhängigkeiten leiten sich wie folgt her:

Definition 4.30 (Kausale Abhängigkeiten im imperativen Modell)**.** Es sei $G = (N, E, \lambda)$ ein Prozessmodell und a und b zwei Aktivitäten dieses Modells. Für das Tupel (a, b) gilt

- $=> (a, b)$, falls entweder b a dominiert oder b a postdominiert
- $== (a, b)$, falls entweder $=> (a, b)$ und a b dominiert sowie a b postdominiert (d. h. $=> (b, a)$) oder a und b verpflichtend parallel sind.
- $>< (a, b)$, falls a und b zum selben XOR-Fragment gehören, aber in diesem zu unterschiedlichen Zweigen, und es weder einen Pfad zwischen a und b noch zwischen b und a gibt.
- $\sim (a, b)$, wenn weder $=> (a, b)$ noch $=> (b, a)$ noch $>< (a, b)$ für a und b festgestellt werden kann.

Die exklusive Existenzabhängigkeit $>< (a, b)$ ist hierbei global zu betrachten; wenn zwei Aktivitäten auf unterschiedlichen Zweigen eines XOR-Fragments liegen, dieses aber in einem XOR-Loop liegt (sich also ein Pfad dazwischen befindet), so sind a und b nicht exklusiv.

Auf diese Weise sind aus den reinen Kontrollflussangaben im Prozessmodell relativ detaillierte Angaben über sowohl ordnungsbasierte als auch kausale Abhängigkeiten zwischen je zwei Aktivitäten bestimmt. Diese erlauben, auch feine Unterschiede zwischen Prozessmodellen in ihrem jeweiligen Verhalten zu erkennen, wobei auch die binären Relationen natürlich Abstraktionen des Verhaltens darstellen. Zusätzlich zu den reinen Kontrollflussabhängigkeiten werden in Abschnitt 4.3.6.3 datenflussbasierte Abhängigkeiten betrachtet. Über den Datenfluss können neue, im reinen Kontrollfluss nicht erfasste Abängigkeiten hinzukommen. Bei imperativen Prozessmodellen ist dies in der Regel weniger der Fall, da hier der Kontrollfluss die datenflussbasierten Abhängigkeiten meist beinhaltet, in deklarativen jedoch schon, d. h., ein Informationsgewinn kann bei Hinzunahme solcher Abhängigkeiten echt vorhanden sein.

4.3.6.3 Datenflussbasierte Abhängigkeiten

Um Informationen aus dem Datenfluss in den Abhängigkeiten zu berücksichtigen, wird für die bestehenden Aktivitätenpaare, die keine exklusive Existenzabhängigkeit aufweisen, überprüft, ob ein Datenfluss zwischen diesen beiden Aktivitäten im Modell gegeben ist. Falls ja, so wird den bereits festgestellten Abhängigkeiten ein Hinweis darauf mittels eines hochstestellten ds hinzugefügt. Lediglich den Symbolen $\sim$ und $-$ kann kein solches Symbol hinzugefügt werden, da hier entweder keine zwingende Existenzabhängigkeit festgestellt wurde oder die angegebene Reihenfolge gar nicht erlaubt ist.

Definition 4.31 (Datenflussbasierte Abhängigkeiten)**.** Es sei $G = (N, E, \lambda)$ ein Prozessmodell, für das bereits ordnende und kausale Abhängigkeiten basierend auf dem Kontrollfluss festgestellt wurden und a und b zwei Aktivitäten dieses Modells. Alle Abhängigkeiten zwischen a und b werden zu datenflussbasierten Abhängigkeiten, falls

$$\lambda_3(a) = \delta = \lambda_3(b)$$

für $\delta \in \mathcal{D}$.

Da im Prozessmodell nicht zwischen eingehenden und ausgehenden Datenobjekten unterschieden wird, kann die Definition der datenflussbasierten Abhängigkeiten nur das Verwenden der Datenobjekte berücksichtigen. Würde zwischen eingehenden und ausgehenden Datenobjekten unterschieden, so müsste die Definition dementsprechend angepasst werden, um die Richtung des jeweiligen Datenflusses zu bestimmen (siehe Baumann, 2017). Diese Anpassung kann jedoch, wie in Abschnitt 3.2.2 geschrieben, leicht vorgenommen werden. Für imperative Prozessmodelle ist eine solche Anpassung, da der Kontrollfluss ja vorhanden ist, nicht unbedingt notwendig.

Beispiel 4.11. Betrachte das Prozessmodell aus Abbildung 4.12. In diesem Modell gelten unter anderem folgende Beziehungen:

- A dominiert B, aber nicht B postdominiert A. Von B gibt es keinen Pfad nach A.
- A dominiert E und E postdominiert A. Von E gibt es keinen Pfad nach A.
- G und H sind verpflichtend parallel.
- Es gibt keine zwei Aktivitäten, die optional parallel sind.

Aus diesen und allen weiteren Informationen lassen sich die Abhängigkeiten, wie sie in Tabelle 4.4 in Matrixform eingetragen sind, herleiten. Da zwischen F und E ein Datenfluss besteht, werden die Abhängigkeitssymbole bei F und E mit einem hochgestellten d versehen. Die Symbole der kausalen Abhängigkeiten sind (immer) an der Diagonale gespiegelt, während für die ordnenden Abhängigkeiten die Reihenfolge per Definition eine Rolle spielt. Zum Beispiel gilt, da A B dominiert, aber B A nicht postdominiert $\nleftarrow (A, B)$ und $\rightarrow (A, B)$, kurz: $\nleftrightarrow (A, B)$ (erste Zeile, zweite Spalte der Matrix). Da es aber keinen Pfad von B nach A gibt, gilt auch $-(B, A)$ (zweite Zeile, erste Spalte der Matrix). Zu den Abhängigkeiten zwischen E und F wird der Datenfluss notiert.

Mit den kontrollfluss- und datenflussbasierten Abhängigkeiten, die sich in ordnende und kausale Flussabhängigkeiten unterscheiden, kann ein zwar abstrahiertes, aber doch recht detailliertes Bild des Verhaltens eines Prozessmodells erstellt werden. Dieses Bild, also die Menge

Tabelle 4.4: Flussabhängigkeiten für das Beispielprozessmodell aus Abbildung 4.12. Die Leserichtung der Abhängigkeiten ist hierbei (Zeile, Spalte).

	A	B	C	D	E	F	G	H
A		$<=$ $\nleftrightarrow$	$<=$ $\nleftrightarrow$	$<=$ $\nleftrightarrow$	$<=$ $\nleftrightarrow$	$<=$ $\nleftrightarrow$	$<=$ $\nleftrightarrow$	$<=$ $\nleftrightarrow$
B	$=>$ $-$		$><$ $-$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
C	$=>$ $-$	$><$ $-$		$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
D	$=>$ $-$	$\sim$ $-$	$\sim$ $-$		$><$ $-$	$><$ $-$	$><$ $-$	$><$ $-$
E	$=>$ $-$	$\sim$ $-$	$\sim$ $-$	$><$ $-$		$\overset{d}{==}$ $\overset{d}{\nleftrightarrow}$	$==$ $\nleftrightarrow$	$==$ $\nleftrightarrow$
F	$=>$ $-$	$\sim$ $-$	$\sim$ $-$	$><$ $-$	$\overset{d}{==}$ $\overset{d}{-}$		$==$ $\nleftrightarrow$	$==$ $\nleftrightarrow$
G	$=>$ $-$	$\sim$ $-$	$\sim$ $-$	$><$ $-$	$==$ $-$	$==$ $-$		$==$ $\leftrightarrow$
H	$=>$ $-$	$\sim$ $-$	$\sim$ $-$	$><$ $-$	$==$ $-$	$==$ $-$	$==$ $\leftrightarrow$	

der binären Relationen zwischen den Aktivitäten, kann für die Berechnung eines Ähnlichkeitswertes verwendet werden, indem die Relationenmengen bzw. die Abhängigkeitsmatrizen zweier Prozessmodelle miteinander verglichen werden. Durch die Überführung des Verhaltens in Flussabhängigkeiten können auch Modelle, die in unterschiedlichen Sprachen modelliert sind, abgeglichen werden (siehe z. B. Abschnitt 4.5.3.4 für deklarative Modelle).

4.3.6.4 Abgleich zweier Prozessmodelle auf Basis von Flussabhängigkeiten

Sollen zwei Prozessmodelle bezüglich ihres Verhaltens abgeglichen werden und wird das Verhalten mittels Flussabhängigkeiten repräsentiert, so werden die Abhängigkeitsmatrizen beider Prozessmodelle miteinander verglichen. Hierfür wird zunächst eine Hierarchie zwischen den Abhängigkeiten festgelegt, um nicht nur Gleichheit und Ungleichheit feststellen zu können, sondern auch Ähnlichkeit, da für zwei Aktivitäten a und b beispielsweise gilt: $== (a, b) \Rightarrow => (a, b)$. Wird im einen Modell $== (a, b)$ festgestellt und im anderen $=> (M(a), M(b))$, so sind beide Abhängigkeiten nicht gleich, aber auch nicht komplett verschieden. Außerdem kann mittels einer solchen Hierarchie festgestellt werden, ob ein Modell im Vergleich zum anderen Modell über- bzw. unterbestimmt ist in dem Sinn, dass ein Modell all das erlaubt, was auch das andere erlaubt, plus einige Möglichkeiten mehr, oder ob sich die beiden Modelle widersprechen. Über- und Unterbestimmtheit bezieht sich auf die Menge aller möglichen Ausführungspfade.

Es bezeichne dep' und dep'' jeweils eine Abhängigkeit vom selben Typ (ordnend oder kausal). Der Ausdruck $dep' \prec dep''$ kennzeichne den Fall, dass Abhängigkeit dep' restriktiver als Abhängigkeit dep'' ist, also dass die Menge aller Ausführungsmöglichkeiten, die durch dep' induziert wird, eine echte Teilmenge der Möglichkeiten ist, die durch dep'' gegeben sind. Die

Relation $\prec$ ist damit transitiv. Mit $dep' \# dep''$ sei der Fall gekennzeichnet, dass die beiden Abhängigkeiten widersprüchlich sind. Um einen Widerspruch in den Abhängigkeiten festzustellen, werden die Ausführungsmöglichkeiten nach Aktivitäten getrennt betrachtet, was im Folgenden näher erläutert wird. Für die Aktivitäten a und b, deren gegebene (ordnende oder kausale) Flussabhängigkeit auf Widersprüchlichkeit überprüft werden soll, sind die Ausführungsmöglichkeiten, die überhaupt zur Verfügung stehen, a, b, ab, und ba, wobei ein einzelnes a hier im Speziellen bedeutet, dass a ausgeführt werden darf, ohne dass davor oder danach b ausgeführt werden muss, und ab bedeutet, dass es möglich ist, in einer Ausführung zuerst a zu erledigen und irgendwann danach b. Es sei a' das Bild von a und b' das Bild von b. Um auf einen Widerspruch in der Abhängigkeit zwischen a und b zu der zwischen a' und b' zu testen, wird die Menge der Ausführungsmöglichkeiten, die nur Ausführungen mit a bzw. a' umfassen, getrennt von der Menge der Ausführungsmöglichkeiten, die nur Ausführungen mit b bzw. b' umfassen, untersucht. Die Menge der Ausführungsmöglichkeiten mit a kann maximal $\{a, ab, ba\}$ sein, die mit b maximal $\{b, ab, ba\}$. Ein Widerspruch der Flussabhängigkeit zwischen a und b zu der zwischen a' und b' liegt dann vor, wenn der Schnitt über die Menge der Ausführungsmöglichkeiten mit a und die Menge der Ausführungsmöglichkeiten mit a' oder der Schnitt über die Mengen mit b bzw. b' leer ist (oder wenn beide Schnitte leer sind), wobei bei der Schnittbildung a' mit a und b' mit b identifiziert wird. Zwei Abhängigkeiten sind nicht vergleichbar, $dep' \circ dep''$, falls keine Hierarchie, aber auch kein Widerspruch festgestellt werden kann. Bei nicht vergleichbaren Abhängigkeiten überschneiden sich die Mengen ihrer Ausführungsmöglichkeiten teilweise, aber in der getrennten Betrachtung nach Aktivitäten sind beide Schnittmengen ebenfalls nicht leer.

Für die kausalen Abhängigkeiten ergeben sich folgende Ausführungsmöglichkeiten:

- $==$ (a, b): $\{ab, ba\}$
- $=>$ (a, b): $\{ab, ba, b\}$
- $<=$ (a, b): $\{ab, ba, a\}$
- $\sim$ (a, b): $\{ab, ba, a, b\}$
- $><$ (a, b): $\{a, b\}$

Der Fall, dass keine der beiden Aktivitäten ausgeführt wird, wird hier nicht beachtet, da hier nur Abhängigkeiten betrachtet werden und nicht bestimmt wird, ob überhaupt eine Aktivität ausgeführt werden muss. Dies wäre die Eigenschaft der Optionalität (siehe Abschnitt 4.3.3), die für jede Aktivität unabhängig von anderen Aktivitäten bestimmt werden kann.

Mit den drei Relationen $\prec$, $\#$ und $\circ$ lassen sich die kausalen Flussabhängigkeiten wie folgt paarweise miteinander vergleichen. Es besteht eine Hierarchie zwischen:

$$== \quad \prec \quad => \quad \prec \quad \sim$$

bzw.

$$== \quad \prec \quad <= \quad \prec \quad \sim$$

Die Abhängigkeiten $=>$ und $<=$ sind nicht vergleichbar, da sich für $=>$ (a, b) und $<=$ (a, b) die Ausführungsmengen teilweise überschneiden und bei getrennter Betrachtung nach beiden Aktivitäten kein leerer Schnitt auftritt: $\{ab, ba\} \cap \{ab, ba, a\} \neq \emptyset$ (nur Möglichkeiten, die a enthalten) und $\{ab, ba, b\} \cap \{ab, ba\} \neq \emptyset$ (nur Möglichkeiten, die b enthalten):

$$=> \circ <=$$

Für die exklusive Abhängigkeit $><$ gelten

$$>< \ \# \ == \text{ und } >< \ \# \ => \text{ bzw. } >< \ \# \ <= .$$

Die Ausführungsmöglichkeiten für $><(a,b)$ und $==(a,b)$ unterscheiden sich komplett voneinander, also erst recht, wenn sie nach Aktivitäten getrennt betrachtet werden. Für einen Vergleich von $><(a,b)$ und $=>(a,b)$ werden die Mengen getrennt betrachtet. Werden nur Ausführungsmöglichkeiten, die a enthalten, berücksichtigt, so gilt $\{a\} \cap \{ab, ba\} = \emptyset$, d. h., es herrscht ein Widerspruch bezüglich Aktivität a. Im Fall $><(a,b)$ kann a nur allein auftreten (nie zusammen mit b), im Fall $=>(a,b)$ kann a nie allein auftreten (es muss zusammen mit b auftreten). Die Begründung für $<=$ ist analog zu der für $=>$ mit vertauschten Rollen von a und b.

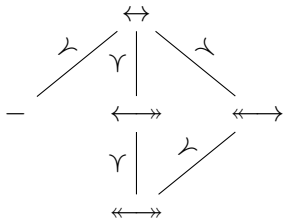
Die Ausführungsmöglichkeiten für $><(a,b)$ sind in denen von $\sim(a,b)$ komplett enthalten, was die Hierarchie

$$>< \ \prec \ \sim$$

zur Folge hat. Für die ordnenden Abhängigkeiten ergeben sich folgende Ausführungsmöglichkeiten:

- $\twoheadleftarrow\!\!\!\!\!\!\rightarrow(a,b)$: $\{ab\}$
- $\leftarrow\!\!\!\!\!\!\rightarrow(a,b)$: $\{ab, b\}$
- $\twoheadleftarrow\!\!\!\!\!\!\rightarrow(a,b)$: $\{ab, a\}$
- $\leftrightarrow(a,b)$: $\{a, b, ab\}$
- $-(a,b)$: $\{a, b\}$

Aus diesen ergeben sich folgende drei parallele Hierarchien:



Des Weiteren gilt der Widerspruch

$$- \ \# \ \twoheadleftarrow\!\!\!\!\!\!\rightarrow,$$

da beide Ausführungsmengen disjunkt sind. Für einen Vergleich von $-$ mit $\leftarrow\!\!\!\!\!\!\rightarrow$ betrachte die Ausführungsmengen nur für a: $\{ab\} \cap \{a\} = \emptyset$. Für einen Vergleich von $-$ mit $\twoheadleftarrow\!\!\!\!\!\!\rightarrow$ betrachte die Ausführungsmengen nur für b: $\{ab\} \cap \{b\} = \emptyset$. Auch hier ergibt sich also ein Widerspruch:

$$- \ \# \ \leftarrow\!\!\!\!\!\!\rightarrow \text{ und } - \ \# \ \twoheadleftarrow\!\!\!\!\!\!\rightarrow$$

Da die Ausführungsmöglichkeit ab in beiden Ausführungsmengen von $\leftarrow\!\!\!\!\!\!\rightarrow$ und $\twoheadleftarrow\!\!\!\!\!\!\rightarrow$ enthalten ist, ist keine Schnittmenge bei getrennter Betrachtung leer, weshalb

$$\leftarrow\!\!\!\!\!\!\rightarrow \ \circ \ \twoheadleftarrow\!\!\!\!\!\!\rightarrow$$

folgt. Zusätzlich zu den drei Relationen $\prec$, # und $\circ$ können zwei Abhängigkeiten natürlich auch gleich sein (=). Für je zwei Abhängigkeitsmatrizen kann so eine Abgleichsmatrix erstellt werden, siehe Beispiel 4.12. Aus dieser lässt sich insbesondere ablesen, ob ein Prozessmodell bezüglich seiner Ausführungsmöglichkeiten im anderen Prozessmodell enthalten ist, also von diesem reproduziert werden kann. Dies kann, unabhängig vom genauen Ähnlichkeitswert, eine wertvolle Information sein.

Beispiel 4.12. Abgeglichen werden das Prozessmodell aus Abbildung 4.12 und das Prozessmodell aus Abbildung 4.14. Für das erste Prozessmodell ist die Abhängigkeitsmatrix in Tabelle 4.4 gegeben, für das zweite Prozessmodell ist die Abhängigkeitsmatrix in Tabelle 4.5 aufgestellt. Ein Vergleich dieser beiden Tabellen, wobei die 1:1-Abbildung die gleich benannten Aktivitäten aufeinander abbildet, liefert die Abgleichsmatrix, wie sie in Tabelle 4.6 gezeigt ist. Die Abgleichsmatrix zeigt, dass in den meisten Paarvergleichen kein Unterschied zwischen den Abhängigkeiten in beiden Modellen besteht. Dort, wo es Unterschiede gibt, ist meist das Modell aus Abbildung 4.12 dasjenige, das eingeschränkteres Verhalten aufweist bzw. das Modell aus Abbildung 4.14 bietet etwas mehr unterschiedliche Ausführungsmöglichkeiten. Ein genauerer Blick offenbart, dass diese Eigenschaft von Aktivität A herrührt. Beim Vergleich der Abhängigkeiten zwischen A und B und zwischen G und H verhält sich die Subsumptionseigenschaft anders herum. Der einzige Widerspruch im Modell besteht zwischen den Aktivitäten A und C. Dadurch, dass die Abgleichsmatrix genau aufzeigt, zwischen welchen zwei Aktivitäten jeweils ein Widerspruch herrscht, kann die Ursache dafür relativ einfach in den Abhängigkeitsmatrizen oder in den Modellen nachvollzogen werden.

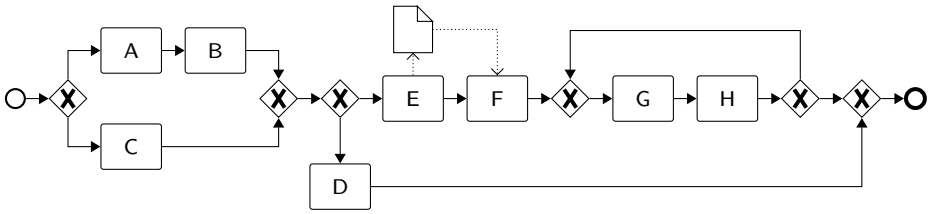


Abbildung 4.14: Beispiel eines imperativen Prozessmodells mit Flussabhängigkeiten.

Nachdem in diesem Abschnitt erklärt wird, wie aus je zwei Abhängigkeitsmatrizen, die das Verhalten der Modelle abstrahieren, mittels einer teilweisen Hierarchiebildung der Flussabhängigkeiten eine Abgleichsmatrix hergeleitet wird und wie aus dieser bereits der Umstand erkannt werden kann, ob ein Modell eine echte Verfeinerung eines anderen Modells darstellt, wird in Abschnitt 4.3.6.5 ein Ähnlichkeitsmaß aus der Abgleichsmatrix definiert. Dieses Maß ist unabhängig davon, ob eine Enthaltenseinsbeziehung vorliegt oder nicht. Allerdings kann für den Fall, dass eine Enthaltenseinsbeziehung zwischen zwei Prozessmodellen besteht, mit diesem Maß der Grad abgelesen werden, welchen Teil des weiter gefassten Modells das strengere Prozessmodell abdeckt.

4.3.6.5 Ähnlichkeitsmaß auf Basis der Abgleichsmatrix

Eine einfache Möglichkeit, um aus der Abgleichsmatrix ein Ähnlichkeitsmaß zu berechnen, ist die, den Abgleichssymbolen der Flussabhängigkeiten einen Wert aus $[0, 1]$ zuzuweisen und darüber zu mitteln. Würde nur das Gleichheitszeichen mit dem Wert 1 und alle anderen Symbole mit dem Wert 0 versehen, so würde dies der Ähnlichkeitsberechnung bei Anwendung der

Tabelle 4.5: Flussabhängigkeiten für das Beispielprozessmodell aus Abbildung 4.14.

	A	B	C	D	E	F	G	H
A		$==$ $\twoheadleftarrow\twoheadrightarrow$	$>\!<$ $-$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
B	$==$ $-$		$>\!<$ $-$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
C	$>\!<$ $-$	$>\!<$ $-$		$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
D	$\sim$ $-$	$\sim$ $-$	$\sim$ $-$		$>\!<$ $-$	$>\!<$ $-$	$>\!<$ $-$	$>\!<$ $-$
E	$\sim$ $-$	$\sim$ $-$	$\sim$ $-$	$>\!<$ $-$		$^{d}==$ $^{d}\twoheadleftarrow\twoheadrightarrow$	$==$ $\twoheadleftarrow\twoheadrightarrow$	$==$ $\twoheadleftarrow\twoheadrightarrow$
F	$\sim$ $-$	$\sim$ $-$	$\sim$ $-$	$>\!<$ $-$	$^{d}==$ $^{d}-$		$==$ $\twoheadleftarrow\twoheadrightarrow$	$==$ $\twoheadleftarrow\twoheadrightarrow$
G	$\sim$ $-$	$\sim$ $-$	$\sim$ $-$	$>\!<$ $-$	$==$ $-$	$==$ $-$		$==$ $\twoheadleftarrow\twoheadrightarrow$
H	$\sim$ $-$	$\sim$ $-$	$\sim$ $-$	$>\!<$ $-$	$==$ $-$	$==$ $-$	$==$ $\leftrightarrow$	

Tabelle 4.6: Abgleichsmatrix der beiden Abhängigkeitsmatrizen aus Tabelle 4.4 und 4.5.

	A	B	C	D	E	F	G	H
A		$\succ$ $\succ$	$\#$ $\#$	$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $\prec$
B	$\succ$ $=$		$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$
C	$\#$ $=$	$=$ $=$		$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$
D	$\prec$ $=$	$=$ $=$	$=$ $=$		$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$
E	$\prec$ $=$	$=$ $=$	$=$ $=$	$=$ $=$		$=$ $=$	$=$ $=$	$=$ $=$
F	$\prec$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$		$=$ $=$	$=$ $=$
G	$\prec$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$		$=$ $\succ$
H	$\prec$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	

kausalen Verhaltensprofile (siehe Abschnitt 2.2.4.5) entsprechen. Um auch die Enthaltenseinsbeziehung oder die sich nicht notwenigerweise widersprechende Nichtvergleichbarkeit zu berücksichtigen, lautet eine differenziertere Wertevergabe beispielsweise wie folgt:

$$= : 1, \prec \, / \, \succ : 0,75, \circ : 0,5 \text{ und } \# : 0.$$

So wird allerdings kein Unterschied gemacht zwischen $\prec$ und $\succ$ und die Distanz in der durch $\prec$ bzw. $\succ$ induzierten Hierarchie wird nicht berücksichtigt. Außerdem werden, wenn über die Anzahl aller Symbole in der Abgleichsmatrix arithmetisch gemittelt wird, die kausalen Abhängigkeiten doppelt so stark gewichtet wie die ordnenden, da für die Bestimmung der kausalen Abhängigkeiten keine Reihenfolge benötigt wird und somit in der oberen Dreiecksmatrix der Abgleichsmatrix alle Information bereits enthalten ist. Dies wird im Weiteren aber absichtlich so belassen, um Kausalität und Reihenfolge bei der Ähnlichkeitsberechnung (Mittelung) gleich stark zu berücksichtigen.

Um die Entfernung in der Hierarchie bei $\prec$ bzw. $\succ$ zu berücksichtigen, muss diese in der Abgleichsmatrix vermerkt sein. Es kennzeichne $\prec/\succ$ einen Hierarchieunterschied von einer Ebene und $\prec\prec/\succ\succ$ einen Hierarchieunterschied von zwei Stufen. Die Abhängigkeiten $==$ und $\sim$ bekommen dann $\prec\prec$ statt wie bisher $\prec$ zugewiesen und werden als weniger ähnlich eingestuft als beispielsweise $==$ und $=>$, denen weiterhin $\prec$ beim Abgleich zugewiesen wird. Tabelle 4.6 zeigt bereits die differenzierte Abgleichsmatrix für die beiden Beispielmodelle, da hier kein $\prec$ / $\succ$ über zwei Ebenen vorliegt. Für diese Differenzierung bei hierarchischen Abhängigkeiten kann eine Wertevergabe folgendermaßen lauten:

$$= : 1, \prec \, / \, \succ : 0,8, \prec\prec \, / \, \succ\succ : 0,6, \circ : 0,5 \text{ und } \# : 0. \tag{4.3}$$

In Tabelle 4.7 ist die Ähnlichkeitswertung für die Abgleichsmatrix aus Tabelle 4.6 gezeigt. Mit dieser Wertung lässt sich ein Ähnlichkeitsmaß wie folgt definieren:

Definition 4.32. Gegeben seien zwei Prozessmodelle G_1 und G_2 mit einer 1:1-Abbildung $M : A_1 \to A_2$. Es bezeichne $c(a, b)$ den Abgleichswert der kausalen Abhängigkeiten zwischen den Aktivitäten $a, b \in A_1$ und $M(a), M(b) \in A_2$, $a \neq b$, und $o(a, b)$ den Abgleichswert der ordnenden Abhängigkeiten zwischen $a, b \in A_1$ und $M(a), M(b) \in A_2$, $a \neq b$, gemäß Gleichung (4.3). Die Ähnlichkeit von G_1 und G_2 bezüglich M unter Berücksichtigung der Flussabhängigkeiten ist

$$VSim_M^{fdep}(G_1, G_2) = \frac{\sum_{(a,b)\in A_1\times A_1, a\neq b} c(a, b) + \sum_{(a,b)\in A_1\times A_1, a\neq b} o(a, b)}{2 \cdot \sum_{(a,b)\in A_1\times A_1, a\neq b} 1} \in [0, 1].$$

Beispiel 4.13. Für die Prozessmodelle aus den Abbildungen 4.12 und 4.14, den zugehörigen Abhängigkeitsmatrizen in Tabellen 4.4 und 4.5, der daraus abgeleiteten Abgleichsmatrix in Tabelle 4.6 und der Wertung wie in Tabelle 4.7 berechnet sich die Verhaltensähnlichkeit auf Basis der Flussabhängigkeiten zu

$$\begin{aligned} VSim_M^{fdep}(G_1, G_2) &= \frac{(2 \cdot 0 + 12 \cdot 0,8 + 42 \cdot 1) + (1 \cdot 0 + 7 \cdot 0,8 + 48 \cdot 1)}{2 \cdot 56} \\ &= \frac{51,6 + 53,6}{2 \cdot 56} \\ &\approx 0,94 \end{aligned}$$

Tabelle 4.7: Wertungsmatrix zur Abgleichsmatrix aus Tabelle 4.6.

	A	B	C	D	E	F	G	H
A		0,8	0	0,8	0,8	0,8	0,8	0,8
		0,8	0	0,8	0,8	0,8	0,8	0,8
B	0,8		1	1	1	1	1	1
	1		1	1	1	1	1	1
C	0	1		1	1	1	1	1
	1	1		1	1	1	1	1
D	0,8	1	1		1	1	1	1
	1	1	1		1	1	1	1
E	0,8	1	1	1		1	1	1
	1	1	1	1		1	1	1
F	0,8	1	1	1	1		1	1
	1	1	1	1	1		1	1
G	0,8	1	1	1	1	1		1
	1	1	1	1	1	1		0,8
H	0,8	1	1	1	1	1	1	
	1	1	1	1	1	1	1	

Eine alternative Berechnung kann zunächst das Vorkommen von $\prec$ und $\succ$ zählen und nur dasjenige Zeichen mit seinem zugewiesenen Ähnlichkeitswert berücksichtigen, das häufiger vorkommt. Das andere, weniger häufige, würde mit 0 bewertet werden, um die gegenläufige Richtung der Enthaltenseinsbeziehung zu bestrafen. Zusätzlich zu den Vergleichen der Abhängigkeiten zwischen je zwei unterschiedlichen Knoten kann für die nach Definition 4.25 bzw. 4.32 leere Diagonale, also für den Vergleich der Beziehung einer Aktivität mit sich selbst mit der Beziehung der abgebildeten Aktivität mit sich selbst, anstatt einer kausalen und einer ordnenden Abhängigkeit die Wiederholbarkeit und die Optionalität bzw. deren Gleichheit oder Ungleichheit aus den Abschnitten 4.3.2 und 4.3.3 eingetragen werden. Diese beiden Merkmale sind in den Flussabhängigkeiten nämlich nicht notwendigerweise berücksichtigt und können somit einen echten Informationsgewinn darstellen. Die Information, ob eine Abhängigkeit rein kontrollflussbasiert ist oder auf einem Datenfluss beruht, ist in der angegebenen Ähnlichkeitsberechnung nicht aufgenommen. Es kann aber, wenn der Ursprung der Flussabhängigkeit (Kontrollfluss oder Datenfluss) bei sonst gleicher Abhängigkeit nicht übereinstimmt, der Wert in der Wertungsmatrix etwas gesenkt werden, von 1 beispielsweise auf 0,9. Da für imperative Prozessmodelle jedoch immer ein Kontrollfluss vorhanden ist und die Datenflussabhängigkeiten in einem zweiten Schritt lediglich durch eine Markierung der vorhandenen Abhängigkeiten ergänzt werden, scheint es in diesem Fall nicht sehr sinnvoll, diese Unterscheidung zu treffen (die Zuordnung der Datenobjekte zu Aktivitäten wird in der datenorientierten Perspektive erfasst). Erst für deklarative Prozessmodelle ist eine solche Unterscheidung angebracht, siehe dazu Abschnitt 4.5.3.4.

Insgesamt lässt sich sagen, dass die Methode, Flussabhängigkeiten zur Ähnlichkeitsbestimmung zu verwenden, eine recht genaue Erfassung des Verhaltens eines Prozessmodells beinhaltet, im Unterschied zur Verwendung der drei Einzelmerkmale Position, Optionalität und Wiederholbarkeit. Dies legt auch der Vergleich mit der Methode der kausalen Verhaltensprofile

nahe, für die die Flussabhängigkeiten eine echte Verfeinerung darstellen. Ihre Bestimmung ist jedoch ungleich aufwändiger als die Bestimmung von Position, Optionalität und Wiederholbarkeit. Außerdem funktioniert die Methode in der gezeigten Version nur für 1:1-Abbildungen, nicht für M:N-Abbildungen. Ein großer Vorteil ist jedoch, dass sie in der gezeigten Form auch auf deklarative Prozessmodelle angewendet werden kann, wie in Abschnitt 4.5.3.4 gezeigt. Besonders für den Abgleich von imperativen und deklarativen Prozessmodellen, bei denen die Aktivitäten als 1:1-Entsprechungen in den zu vergleichenden Modellen auftauchen – falls beispielsweise geklärt werden soll, ob das Verhalten eines deklarativen Modells dem eines imperativen Modells nicht widerspricht – kann die Methode gut eingesetzt werden.

Nachdem für die verschiedenen Prozessperspektiven (Beschreibung der Aktivitäten, Ressourcenzuweisung, Verhalten) Ähnlichkeitsmaße definiert sind, wird in Abschnitt 4.4 der vierstufige Ansatz auf multiperspektivische Modelle unter einem M:N-Abgleich ausgeweitet. Der vierstufige Ansatz gibt die Abbildung aus, die einen größtmöglichen Ähnlichkeitswert liefert, wobei für diesen Ähnlichkeitswert die einzelnen Perspektivenähnlichkeiten miteinander kombiniert werden.

4.4 Berechnung eines globalen Ähnlichkeitswertes für M:N-Abbildungen auf multiperspektivischen Prozessmodellen

Analog zum Vorgehen aus Abschnitt 2.2.1, bei dem in vier Schritten ein Ähnlichhkeitswert auf Basis von 1:1-Abbildungen über eine Mittelung der drei einzelnen Ähnlichkeitswerte $fsim_M$ (Ähnlichkeit der Aktivitätenbeschreibungen, des Verhaltens oder der Struktur), $fsubn_M$ (Anteil der abgebildeten Knoten) und $fsube_M$ (Anteil der abgebildeten Kanten) berechnet wird, soll auch bei der Verwendung von M:N-Abbildungen und unter Berücksichtigung weiterer Perspektivenähnlichkeiten verfahren werden. Die Übernahme dieses Vorgehens wird von Baumann et al. (2014) beschrieben. Das heißt, dass sich die Berechnung eines Ähnlichkeitswerts wieder in vier Schritte unterteilt, die in diesem Fall sind:

1. Festlegen einer M:N-Abbildung der Aktivitäten.

2. Berechnung der Ähnlichkeit der einzelnen Prozessperspektiven auf Basis der festgelegten Abbildung.

3. Kombination (Mittelung) der im vorigen Schritt berechneten Einzelähnlichkeiten unter Berücksichtigung der gelöschten Knoten und Kanten.

4. Maximierung des Ähnlichkeitswerts durch Finden einer optimalen Abbildung.

4.4.1 Schritt 1: Festlegen einer M:N-Abbildung

Das Vorgehen in Schritt 1 bedarf keiner genaueren Erläuterung. Ein Prozessmodell G_1 wird auf das zu vergleichende Modell G_2 mittels der M:N-Abbildung $M : G_1 \rightarrow G_2$ gemäß Definition 3.3 abgebildet. Hieraus ergeben sich die Partitionen P_1 bzw. P_2 der Aktivitätenmengen der beiden Modelle, auf Basis derer in Schritt 2 eine Ähnlichkeit berechnet wird.

4.4.2 Schritt 2: Berechnen der Perspektivenähnlichkeiten

Die Ähnlichkeitsberechnungen in Schritt 2 erfolgen nun für alle (verfügbaren) Perspektiven des Prozessmodells, um so viel Information wie möglich aus den Modellen beim Ähnlichkeitsabgleich zu berücksichtigen. Die Funktion $BSim_M(G_1, G_2)$ (Definition 4.2) gibt die Ähnlichkeit der funktionalen Perspektive an. Mit $ASim_M(G_1, G_2)$ (Definition 4.7) wird die Ähnlichkeit der organisatorischen Perspektive berechnet. Genau so wird mittels $DSim_M(G_1, G_2)$ (Definition 4.9) und $SSim_M(G_1, G_2)$ (Definition 4.11) die Ähnlichkeit der datenorientierten bzw. der operationalen Perspektive bestimmt. Die Ähnlichhkeit der verhaltensorientierten Perspektive kann beispielsweise mit den drei unabhängig voneinander zu bestimmenden Teilperspektiven der Position, der Wiederholbarkeit und der Optionalität gemessen werden. Hier gehen also $VSim^{\pi}_M(G_1, G_2)$ (Definition 4.14), $VSim^{\rho}_M(G_1, G_2)$ (Definition 4.17) und $VSim^{o}_M(G_1, G_2)$ (Definition 4.20) in den nachfolgenden Schritt 3 ein. Alternativ kann für die Verhaltensperspektive auch $VSim^{fdep}_M(G_1, G_2)$ (Definition 4.32), wenn nur 1:1-Abbildungen in Schritt 1 zugelassen sind, erfasst werden oder $VSim^{ord}_M(G_1, G_2)$ (Definition 4.24), wenn keine Schleifen im Prozessmodell auftauchen. Auch eine (beliebige) Kombination mehrerer oder aller genannter Verhaltensähnlickeiten kann erfolgen, wobei es dann der Fall sein kann, dass bestimmte Aspekte der Prozessmodelle in mehreren Maßen berücksichtigt werden und somit bei der Verrechnung in Schritt 3 implizit öfter auftauchen.

Um ungeeignete Abbildungskandidaten möglichst früh aus der Menge der möglichen Abbildungen zu entfernen, können einzelne Perspektivenähnlichkeiten auch als Ausschlusskriterium fungieren, wie von Baumann et al. (2014) vorgeschlagen. So ist es beispielsweise vorstellbar, Abbildungen, die einen Ähnlichkeitswert von 0 in der Datenperspektive liefern, sofort zu verwerfen, ohne die Ähnlichkeitswerte der übrigen Perspektiven zu betrachten. Die Datenperspektive wird hier deshalb angeführt, weil die Zuweisung von IDs zu einzelnen Objekten, ohne eine Zwischenschicht von Rollen und Gruppen wie bei der organisatorischen Perspektive, klar geregelt ist. Es kann aber auch jede weitere Perspektive für solch ein Ausschlusskriterium herangezogen werden, vorausgesetzt, das Modell bildet die jeweilige Prozessperspektive in ausreichender Qualität ab. Es ist sogar möglich, das Ausschlusskriterium einen Schritt früher auf die Ähnlichkeit der Abbildungselemente anzuwenden. Wenn in der Abbildung ein Paar $(p, M(p))$ mit $Sim(p, M(p)) = 0$ enthalten ist für ein ausgewähltes Ähnlichkeitsmaß Sim, so wird die Abbildung nicht weiter betrachtet. Auch eine Kombination aus solchen Nullwerten zum Ausschluss der jeweiligen Abbildung ist denkbar.

4.4.3 Schritt 3: Mittelung der Perspektivenähnlichkeiten mit Anteil abgebildeter Knoten/Kanten

In Schritt 3 werden die in Schritt 2 bestimmten Ähnlichkeiten der einzelnen Perspektiven miteinander über ein gewichtetes Mittel kombiniert. Wie auch schon in Abschnitt 2.2.1.3 werden durch Einbeziehen des Anteils der gelöschten Knoten und Kanten, bzw. der abgebildeten Knoten und Kanten, diejenigen Abbildungen, die einen guten Ausgleich aus gelöschten und abgebildeten Elementen finden, bevorzugt. Je weniger Knoten abgebildet werden, desto höher kann tendenziell der Ähnlichkeitswert dieser Knoten werden, da nur „die ähnlichsten“ Knotenmengen tatsächlich aufeinander abgebildet werden. Die Auswahl weniger abgebildeter Knoten führt allerdings zu einem geringen Wert des Anteils der abgebildeten Knoten und Kanten, der bei der Verrechnung den globalen Ähnlichkeitswert somit verringert. Die Berechnung des globalen Ähnlichkeitswerts erfolgt wie in Definition 4.33 als gewichtetes Mittel der einzelnen Ähnlichhkeiten.

Definition 4.33 (Ähnlichkeit multiperspektivischer Prozessmodelle auf Basis einer Abbildung M)**.** Für zwei Prozessmodelle G_1 und G_2 mit einer Beschreibungsähnlichkeit $BSim_M$, einer Agentenähnlichkeit $ASim_M$, einer Dokumentenähnlichkeit $DSim_M$, einer Werkzeugähnlichkeit $SSim_M$, einer Verhaltensähnlichkeit $VSim_M$, dem Anteil abgebildeter Knoten $NSim_M$ und dem Anteil abgebildeter Kanten $ESim_M$ errechnet sich deren Ähnlichkeit unter einer M:N-Abbildung $M : P_1 \to P_2$, wobei P_1 eine Partition von A_1 und P_2 eine Partition von A_2 ist, über

$$\begin{aligned} GSim_M(G_1, G_2) = {} & w_1 \cdot BSim_M(G_1, G_2) + w_2 \cdot VSim_M(G_1, G_2) \\ & + w_3 \cdot ASim_M(G_1, G_2) + w_4 \cdot DSim_M(G_1, G_2) + w_5 \cdot SSim_M(G_1, G_2) \\ & + w_6 \cdot NSim_M(G_1, G_2) + w_7 \cdot ESim_M(G_1, G_2) \end{aligned}$$

mit $\sum_{i=1,\ldots,7} w_i = 1$, $w_i \geq 0$.

Für $VSim_M(G_1, G_2)$ in Definition 4.33 kann beispielsweise

$$\begin{aligned} & VSim_M(G_1, G_2) \\ & = w_{2,\pi} \cdot VSim^{\pi}_M(G_1, G_2) + w_{2,\rho} \cdot VSim^{\rho}_M(G_1, G_2) + w_{2,o} \cdot VSim^{o}_M(G_1, G_2) \end{aligned}$$

mit $w_{2,\pi} + w_{2,\rho} + w_{2,o} = 1$ und $w_{2,\cdot} \geq 0$ eingesetzt werden. In der Berechnung der Ähnlichkeit $GSim_M(G_1, G_2)$ hätte dann $VSim^{\pi}_M(G_1, G_2)$ mit den angegebenen Gewichten einen Anteil von $w_2 \cdot w_{2,\pi}$. Die Gewichtung kann dabei grundsätzlich individuell angepasst werden, je nachdem, welche Perspektive stärker oder weniger stark in die Ähnlichkeitsberechnung eingehen soll.

Zu definieren ist nun noch, wie der Anteil der abgebildeten Knoten $NSim$ und der Anteil der abgebildeten Kanten $ESim$ unter einer M:N-Abbildung bestimmt wird. Wie in der Definition der M:N-Abbildung (Definition 3.3) angegeben, werden die Aktivitäten von G_1 gelöscht, die auf die leere Menge abgebildet werden, und die aus G_2 gelöscht, die das Bild der leeren Menge sind.

Definition 4.34 (Anteil gelöschter bzw. abgebildeter Aktivitäten unter einer M:N-Abbildung)**.** Es seien G_1 und G_2 zwei Prozessmodelle und $M : P_1 \to P_2$ eine bijektive M:N-Abbildung gemäß Definition 3.3. Es seien, falls vorhanden, $p_1 \subseteq A_1$ mit $M(p_1) = \emptyset$ und $p_2 \subseteq A_2$ mit $M(\emptyset) = p_2$. Der Anteil der gelöschten Knoten ist damit $\frac{|p_1|+|p_2|}{|A_1|+|A_2|}$, was zu einem Anteil abgebildeter Knoten von

$$NSim_M(G_1, G_2) = 1 - \frac{|p_1| + |p_2|}{|A_1| + |A_2|}$$

führt.

Für den Anteil abgebildeter bzw. gelöschter Kanten lassen sich zwei mögliche Definitionen finden, die jedoch eines gemeinsam haben: Nicht jede Kante im ursprünglichen Modell kann sinnvollerweise bei der Berechnung des Kantenanteils berücksichtigt werden. Dadurch, dass Aktivitäten zu Mengen zusammengefasst werden, kann es passieren, dass sich in einer solchen Menge zwei Aktivitäten befinden, die im ursprünglichen Modell mit einer Kante verbunden sind. Dadurch, dass diese Aktivitäten nun in einer Menge sind und nur die Menge als Ganzes abgebildet wird, spielen solche Kanten für diese Abbildung keine Rolle. Diese Kanten werden als neutrale Kanten bezeichnet. Beachte, dass vor der Betrachtung der Kanten die Gateways

aus den Prozessmodellen wie bei der Bestimmung der relativen Position der Aktivitäten (siehe Abschnitt 4.3.1) abstrahiert bzw. ausgeblendet werden. Ein Prozessmodell hat damit nur noch ein Start- und ein Endereignis mit potentiell mehr als einer aus- bzw. eingehenden Kante und Aktivitäten, die direkt miteinander verbunden sind und mehrere ein- und ausgehende Kanten haben können.

Definition 4.35 (Neutrale Kanten bezüglich einer Partition)**.** Es seien G_1 und G_2 zwei Prozessmodelle, auf denen eine Abbildung $M : P_1 \to P_2$ jeweils eine Partition der Aktivitätenmengen A_1 und A_2 induziert. Eine Kante $e_j \in E_i$, $i = 1, 2$, wird als neutrale Kante bezeichnet, falls entweder

$$e_j = (a_{j,1}, a_{j,2}) \text{ mit } a_{j,1}, a_{j,2} \in p_j \subseteq P_i,\ a_{j,1}, a_{j,2} \in A_i$$

oder

$$e_j = (e_{i,start}, a_j) \text{ bzw. } e_j = (a_j, e_{i,end}),\ a_j \in A_i, e_{i,start}, e_{i,end} \in N_i.$$

Die Menge der relevanten Kanten E^*_M ist dann die Menge aller Kanten ohne die neutralen Kanten:

$$E^*_{M,i} = \{e_j \in E_i \mid e_j \text{ ist nicht neutral}\}$$

Es sind also nur solche Kanten relevant, die Aktivitäten in verschiedenen Mengen miteinander verknüpfen. Auch Kanten, die vom Startereignis ausgehen oder zum Endereignis führen, sind neutral, da Start- und Endereignisse nicht explizit über M abgebildet werden. Die relevanten Kanten lassen sich, analog zu Definition 2.7, in abgebildete und gelöschte Kanten einteilen, wobei hier zwischen zwei Möglichkeiten, wie diese Einteilung genau erfolgen kann, gewählt werden kann.

Die erste Möglichkeit berücksichtigt alle Kanten, außer den neutralen, die im ursprünglichen Modell vorhanden sind und zählt diese einzeln. Die zweite Möglichkeit überprüft, ob im ursprünglichen Modell zwischen zwei beliebigen Aktivitäten, die durch die Abbildung in unterschiedlichen Mengen liegen, eine Kante existiert oder nicht. Es wird also für zwei Aktivitätenmengen lediglich entschieden, ob eine Kante existiert und nicht, wie viele es im ursprünglichen Modell sind. Da Kanten gerichtet sind, kann es für zwei Aktivitätenmengen mit der zweiten Methode maximal eine Kante in die eine und eine Kante in die andere Richtung geben.

Definition 4.36 (Abgebildete und gelöschte Kanten bei einzelner Zählung)**.** Es seien G_1 und G_2 zwei Prozessmodelle mit Abbildung $M : P_1 \to P_2$ und relevanten Kantenmengen $E^*_{M,1}$ und $E^*_{M,2}$. Es sind $p_1, q_1 \in P_1 \setminus \emptyset$ mit $p_1 \neq q_1$ und $p_2, q_2 \in P_2 \setminus \emptyset$ mit $p_2 \neq q_2$. Die Menge der abgebildeten Kanten bei einzelner Zählung ist gegeben als

$$Sube'_M(G_1, G_2) = \{(a_1, b_1) \in E^*_{M,1} \cup (a_2, b_2) \in E^*_{M,2} \mid a_1 \in p_1, b_1 \in q_1, a_2 \in p_2, b_2 \in q_2, M(p_1) = p_2, M(q_1) = q_2\}$$

Die Menge der gelöschten Kanten ist dann $Skipe'_M(G_1, G_2) = (E^*_{M,1} \cup E^*_{M,2}) \setminus Sube'_M$.

Für die zweite Möglichkeit der Kantenzählung wird der Begriff der abstrahierten Kante, einer Kante zwischen zwei Aktivitätenmengen (nicht zwischen zwei Aktivitäten), eingeführt.

Definition 4.37 (Abstrahierte Kanten bezüglich einer Partition)**.** Es sei G ein Prozessmodell und P eine disjunkte, vollständige Partitionierung der Aktivitäten von G. Des Weiteren sei E^*_M die Menge der relevanten Kanten auf Basis von P. Die Menge der abstrahierten Kanten $\tilde{E}_M$ ist:

$$\tilde{E}_M = \{\tilde{e} = (p, q) \mid \exists a \in p, b \in q, p \neq q : (a, b) \in E^*_M\}$$

Beachte, dass $(p, q) \in \tilde{E}_M$ nicht dasselbe ist wie $(q, p) \in \tilde{E}_M$. Es kann zwischen den Aktivitätenmengen p und q also bis zu zwei abstrahierte Kanten geben: von p nach q und von q nach p.

Definition 4.38 (Abgebildete und gelöschte Kanten bei abstrahierter Zählung)**.** Es seien G_1 und G_2 zwei Prozessmodelle mit Abbildung $M : P_1 \to P_2$ und abstrahierten Kantenmengen $\tilde{E}_{M,1}$ und $\tilde{E}_{M,2}$. Es sind $p_1, q_1 \in P_1 \setminus \emptyset$ mit $p_1 \neq q_1$ und $p_2, q_2 \in P_2 \setminus \emptyset$ mit $p_2 \neq q_2$. Die Menge der abgebildeten Kanten bei abstrahierter Zählung ist gegeben als

$$Sube''_M(G_1, G_2) = \{(p_1, q_1) \in \tilde{E}_1 \cup (p_2, q_2) \in \tilde{E}_2 \mid M(p_1) = p_2, M(q_1) = q_2\}$$

Die Menge der gelöschten Kanten ist dann analog zu oben $Skipe''_M(G_1, G_2) = (\tilde{E}_{M,1} \cup \tilde{E}_{M,2}) \setminus Sube''_M(G_1, G_2)$.

Die zweite Definition von abgebildeten bzw. gelöschten Kanten (Definition 4.38) ist losgelöst von der Betrachtung einzelner Aktivitäten und Kanten zwischen diesen. Sie trägt somit der Sichtweise von Aktivitätenmengen als Bestandteil der Prozessmodelle, nicht von einzelnen Aktivitäten, besser Rechnung. Es gilt dabei immer

$$|Sube''_M(\cdot, \cdot)| \leq |Sube'_M(\cdot, \cdot)|.$$

Beispiel 4.14 illustriert den Unterschied der beiden Definitionen von abgebildeten bzw. gelöschten Kanten.

Beispiel 4.14. Gegeben sind die beiden Prozessmodelle G_1 und G_2 aus Abbildung 4.15 mit der angedeuteten Abbildung. Die Modelle sind dieselben wie in Abbildung 3.4, lediglich mit ausgeblendeten Gateways. Die neutralen Kanten sind gestrichelt eingezeichnet.

- Neutrale Kanten in G_1 sind $(e_{1,start}, n_{1,1}), (n_{1,1}, n_{1,2}), (n_{1,1}, n_{1,3}), (n_{1,4}, n_{1,5})$ und die beiden einzigen relevanten Kanten sind $(n_{1,2}, n_{1,4})$ und $(n_{1,3}, n_{1,4})$.
- Neutrale Kanten in G_2 sind $(e_{2,start}, n_{2,1}), (e_{2,start}, n_{2,3}), (n_{2,1}, n_{2,2})$ und die beiden einzigen relevanten Kanten sind $(n_{2,2}, n_{2,4})$ und $(n_{2,3}, n_{2,4})$.
- Die Menge der unter M abgebildeten Kanten (einzeln gezählt) ist $Sube'_M(G_1, G_2) = \{(n_{1,2}, n_{1,4}), (n_{1,3}, n_{1,4}), (n_{2,2}, n_{2,4})\}$.
  Die Kante $(n_{2,3}, n_{2,4})$ ist die einzige gelöschte Kante.

Mit $p_1 = \{n_{1,1}, n_{1,2}, n_{1,3}\}, q_1 = \{n_{1,4}, n_{1,5}\}, p_2 = \{n_{2,1}, n_{2,2}\}, q_2 = \{n_{2,4}\}$ und $r_2 = \{n_{2,3}\}$ ergibt sich für die abstrahierten Kanten:

- $\tilde{E}_1 = \{(p_1, q_1)\}$
- $\tilde{E}_2 = \{(p_2, q_2), (r_2, q_2)\}$
- Die Menge der unter M abgebildeten, abstrahierten Kanten ist $Sube''_M(G_1, G_2) = \{(p_1, q_1), (p_2, q_2)\}$.
  Die Kante (r_2, q_2) ist die einzige gelöschte Kante.

Mit der Menge der abgebildeten Kanten lässt sich nun der Anteil der abgebildeten Kanten bestimmen. Für die nachfolgende Definition wird die abstrahierte Sichtweise auf Kanten verwendet.

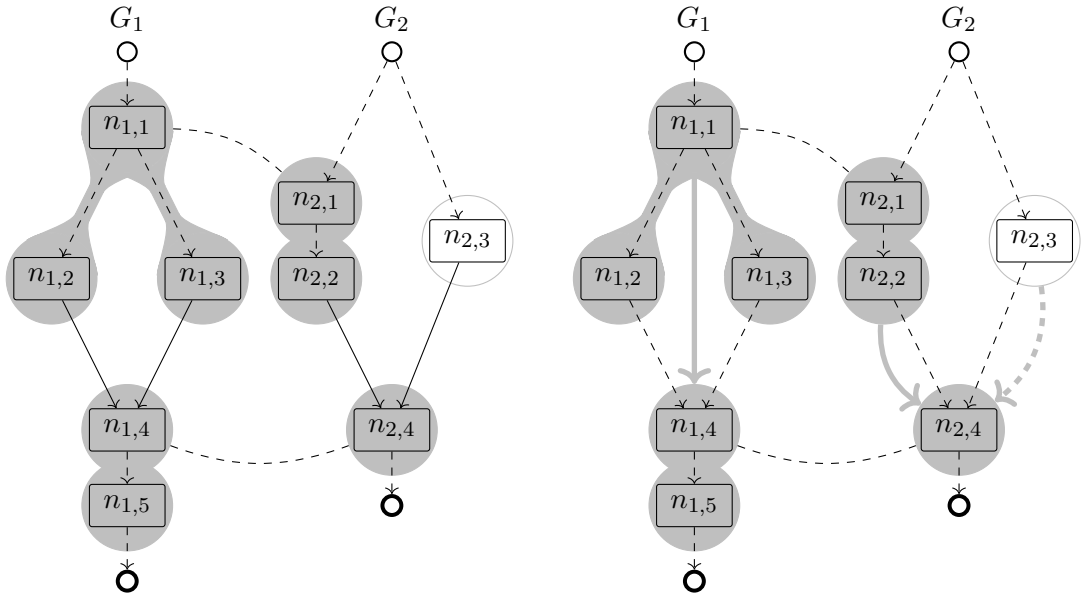


Abbildung 4.15: Beispielmodelle mit ausgeblendeten Gateways und Abbildung. Links alle Kanten einzeln (neutrale Kanten gestrichelt, relevanten Kanten durchgezogen), rechts die abstrahierten Kanten (graue Pfeile; abgebildete Kanten durchgezogen, gelöschte Kanten gestrichelt).

Definition 4.39 (Anteil gelöschter bzw. abgebildeter Kanten unter einer M:N-Abbildung)**.** Es seien G_1 und G_2 zwei Prozessmodelle und $M : P_1 \to P_2$ eine bijektive M:N-Abbildung gemäß Definition 3.3. Es seien $\tilde{E}_{M,1}$ und $\tilde{E}_{M,2}$ die Mengen der abstrahierten Kanten der beiden Prozessmodelle. Der Anteil der abgebildeten Kanten ist

$$ESim_M(G_1, G_2) = \frac{|Sube''_M(G_1, G_2)|}{|\tilde{E}_{M,1}| + |\tilde{E}_{M,2}|}$$

für $|\tilde{E}_{M,1}| + |\tilde{E}_{M,2}| > 0$. Falls $|\tilde{E}_{M,1}| + |\tilde{E}_{M,2}| = 0$, dann setze $ESim_M(G_1, G_2) = 1$. Der Anteil der gelöschten Kanten ist $\frac{|Skipe''_M(G_1,G_2)|}{|\tilde{E}_{M,1}|+|\tilde{E}_{M,2}|} = 1 - ESim_M(G_1, G_2)$.

Da der Anteil der abgebildeten Kanten genau wie der Anteil der abgebildeten Knoten primär dazu dient zu verhindern, dass zu viele Modellelemente gelöscht und nur perfekte Matches von Aktivitäten zugelassen werden, wird der Fall, dass keine Kante gelöscht wird, weil keine zu löschende Kante existiert, also dass sowohl $\tilde{E}_{M,1}$ und $\tilde{E}_{M,2}$ leer sind, mit einer Ähnlichkeit von 1 bewertet. Da nichts gelöscht wird, soll auch nichts bestraft werden.

4.4.4 Schritt 4: Maximieren des Ähnlichkeitswerts

Die Ähnlichkeit zweier Prozessmodelle G_1 und G_2 ist der größte Wert von $GSim_M(G_1, G_2)$, wenn alle möglichen Abbildungen M betrachtet werden.

Definition 4.40 (Globale Ähnlichkeit auf Basis von M:N-Abbildungen)**.** Die globale Ähnlichkeit $GSim$ der beiden Prozessmodelle G_1 und G_2 erhält man über

$$GSim(G_1, G_2) = \max_M GSim_M(G_1, G_2).$$

Hierbei ist die optimale[2] M:N-Abbildung M^* gegeben durch

$$M^* = \underset{M:P_1 \to P_2}{\operatorname{argmax}} GSim_M(G_1, G_2),$$

wobei P_1 und P_2 die von M induzierten Partitionen sind.

Dieses Optimierungsproblem ist nicht einfach zu lösen, da die Anzahl unterschiedlicher M:N-Abbildungen hyperexponentiell mit der Anzahl der Aktivitäten in den Modellen wächst. Die genaue Wachstumsvorschrift kann in Anhang A.3 nachgelesen werden. Die prototypische Implementierung in Kapitel 5 zeigt eine Möglichhkeit, wie mit Hilfe ganzzahliger, linearer Optimierung ein optimales M gefunden werden kann.

Wie in diesem Abschnitt gesehen, ist das Verfahren zum Finden der besten Abbildung für den multiperspektivischen M:N-Ansatz in Grundzügen derselbe, wie er für 1:1-Abbildungen in der Literatur genannt ist (Abschnitt 2.2.1). Die größten Anpassungen müssen bei der Berechnung des Anteils der abgebildeten Kanten vorgenommen werden und die zu maximierende Zielfunktion an sich ist um die hinzugenommenen Perspektiven erweitert. Durch die individuell anpassbare Gewichtung der Einzelähnlichkeiten und auch durch die Möglichkeit, neue Einzelähnlichkeiten einfach mit in die Zielfunktion aufzunehmen (oder auch nicht geeignete wegzulassen), bietet der vierstufige Ansatz eine große Bandbreite an Einsatzmöglichkeiten. Der modulare Aufbau – Wahl der Einzelähnlichkeiten, Wahl der zugrunde liegenden Abbildung, Wahl der Zielfunktion – steht in Verbindung mit dem in Abschnitt 3.6 genannten Vorgehen bei der Auswahl der Abgleichsmethoden. Die Modellart spielt für den vierstufigen Ansatz nicht direkt eine Rolle (diese ist ja von vornherein vorgegeben), die Wahl der Abbildung und der Perspektiven jedoch schon. Die Abbildungsart und die verwendeten Perspektiven tauchen im vierstufigen Ansatz an den entsprechenden Stellen auf, d. h., der vierstufige Ansatz spiegelt genau das konzeptuelle Vorgehen wider. Außer der Wahl der Abbildung im ersten Schritt und der Wahl der Perspektiven und deren Ähnlichkeitsberechnung im zweiten Schritt ist auch im dritten Schritt eine Wahlmöglichkeit durch eine andere Art der Zielfunktion gegeben. Die Zielfunktion ist in Abschnitt 3.6 nicht genannt, da sie mit den Prozessmodellen an sich nicht in Verbindung steht. Der vierte Schritt, das Maximieren der Zielfunktion, bietet konzeptuell keine Stellschraube, allerdings ist die Umsetzung des Optimierungsproblems, vor allem wenn M:N-Abbildungen betrachtet werden, keine triviale Angelegenheit. Insgesamt ergeben sich für den in dieser Arbeit geschilderten Modellabgleich folgende fünf Module, die unabhängig voneinader gewählt bzw. bestimmt werden können:

1. Modellart: Ist von außen vorgegeben.
2. Prozessperspektiven: Obermenge ist von außen vorgegeben; Einschränkungen sind möglich.
3. Abbildung: Ist zu wählen; M:N-Abbildung ist die allgemeinste Form.
4. Zielfunktion: Gewichtung ist zu wählen; prinzipiell ist auch eine andere Funktion als ein gewichtetes Mittel vorstellbar.
5. Optimierung: Geeigneter Algorithmus ist zu wählen.

[2]Genauer gesagt, liefert argmax wieder eine Menge bester Abbildungen. Ist diese Menge mehrelementig, so wird ein beliebiges Element daraus als M^* ausgezeichnet.

Von Weidlich et al. (2010a) wird eine andere Möglichkeit, einen Abgleich durchzuführen, beschrieben, der nicht den vierstufigen Ansatz, wie er in dieser Arbeit gezeigt wird, verwendet. Das Vorgehen von Weidlich et al. (2010a) ist das Folgende: In einem ersten Schritt werden Aktivitäten eines Prozessmodells zu Gruppen zusammengefügt, wobei diese Gruppen den in Abschnitt 3.3.1 genannten Bedingungen genügen müssen. In einem zweiten Schritt werden potentielle Matches gesucht, das heißt, es wird ein Ähnlichkeitswert für alle Kombinationen aus zwei einzelnen Aktivitäten (1:1) oder aus einer einzelnen Aktivität und einer Aktivitätengruppe (1:N) berechnet. Alle Paare, deren Ähnlichkeitswert über einem festgelegten Schwellenwert liegt, gelten als potentielle Matches. Diese können sich natürlich überschneiden. Für die Ähnlichkeitsberechnung werden vier verschiedene Möglichkeiten vorgeschlagen, wobei sich alle auf die Beschriftungen der Aktivitäten sowie der Beschriftungen der Rollen und Datenobjekte beziehen. In einem dritten Schritt werden die Werte der gefundenen potentiellen Matches angepasst. Dies geschieht über sogenannte Booster, die verschiedene Eigenschaften der gefundenen Matches überprüfen und gegebenenfalls einige der potentiellen Matches bevorzugen bzw. andere vernachlässigen. So werden beispielsweise Matches bevorzugt, die einen anderen Match komplett enthalten. Genauso werden Matches, die bestimmte strukturelle Eigenschaften erfüllen, bevorzugt. Im vierten Schritt werden mittels der angepassten Werte diejenigen Matches ausgewählt, sodass diese Menge an Matches eine gültige Abbildung ergibt, d. h. insbesondere, dass sich die verwendeten Aktivitäten nicht überschneiden. Der verwendete Algorithmus ist ein Greedy-Algorithmus, der meist gute Ergebnisse liefert, aber nicht erschöpfend und somit nicht unbedingt optimal ist. Eine Erweiterung um eine 1-look-ahead-Strategie bei Wertegleichheit ist ebenfalls angegeben. Außerdem können Abbildungen mittels sogenannter Evaluatoren bewertet werden. Einer davon ist der Graph Edit Distance Evaluator. Dieser berechnet den Editierabstand der beiden verglichenen Modelle auf Basis der gefundenen Abbildung und gibt somit ein Qualitätskriterium der Abbildung an. Das heißt, es kann passieren, dass die Abbildung zwar gute Ähnlichkeitswerte bezogen auf die Beschriftungen der Aktivitäten findet, ein Vergleich der Struktur der beiden Modelle jedoch kaum Ähnlichkeit aufweist. Im vierstufigen Ansatz, wie er in der vorliegenden Arbeit gezeigt wird, wird die Struktur bzw. das Verhalten der Modelle gleich von vornherein mit betrachtet, um nicht erst am Schluss der Berechnung festzustellen, dass die gefundene Abbildung unzureichend ist. Das von Weidlich et al. (2010a) gezeigte Vorgehen ist iterativ, d. h., bei einem schlechten Wert des Evaluators wird eine neue Abbildung gesucht.

Ein Vergleich des vierstufigen Ansatzes und des iterativen Ansatzes von Weidlich et al. (2010a) kommt zu dem Schluss, dass falls der Greedy-Algorithmus auf Anhieb eine Abbildung findet, die vom gewählten Evaluator als gut genug bewertet wird, was wieder über einen vorher festgelegten Schwellenwert geschieht, der iterative Ansatz deutlich schneller abläuft als der vierstufige Ansatz. Der Optimierungsschritt des vierstufigen Ansatzes ist in der vorliegenden Arbeit mittels ganzzahliger linearer Optimierung implementiert (Abschnitt 5.2.3), was im Unterschied zum Greedy-Algorithmus ein erschöpfendes Verfahren ist und eine tatsächlich beste Abbildung liefert. Läuft der iterative Ansatz jedoch mehrmals durch, weil keine der zunächst gefundenen Abbildungen das Qualitätskriterium des Evaluators erfüllt, so ist der Laufzeitvorteil dahin. Selbst wenn nach mehreren Durchläufen eine Abbildung gefunden wird, die gut genug ist, ist diese nicht zwangsläufig die beste bzw. eine der besten. Wenn die Anwendung es nicht verlangt, unbedingt die beste Abbildung zu finden, sondern es ausreicht, eine minimale Ähnlichkeit für beispielsweise Clusterzwecke zu erfüllen, ist der Greedy-Ansatz jedoch ausreichend. Der Optimierungsschritt im vierstufigen Ansatz kann ebenfalls mittels eines Greedy-Algorithmus durchgeführt werden (Dijkman et al., 2009b). Der von Weidlich

et al. (2010a) vorgeschlagene Ansatz unterstützt, wie oben geschrieben, 1:N-Abbildungen, allerdings keine M:N-Korrespondenzen. Hier ist eine Anpassung der Abbildungsdefinition auf eine solche, wie sie im vierstufigen Ansatz verwendet wird, notwendig.

4.5 Abgleich von deklarativen Prozessmodellen

Im Gegensatz zu imperativen Prozessmodellen, die, da sie eher strikte Routineprozesse beschreiben, einen eindeutig vorgeschriebenen und vorab genau spezifizierbaren Kontrollfluss haben, dienen deklarative Prozessmodelle der Beschreibung von flexiblen, entscheidungsintensiven Prozessen mit einem sich dynmanisch entwickelnden, vorab kaum spezifizierbaren Kontrollfluss (Schönig, 2015). Dies lässt darauf schließen, dass sich imperative und deklarative Prozessmodelle vor allem bei der Formulierung der verhaltensorientierten Perspektive voneinander unterscheiden (Fahland et al., 2010). Deswegen werden in diesem Abschnitt, nach einem kurzen Überblick über die in der verwandten Literatur genannten Abgleichsmöglichkeiten, zunächst die funktionale Perspektive und die drei Ressourcenperspektiven (organisatorisch, datenorientiert, operational) auf Abgleichsmethoden hin untersucht, bevor die verhaltensorientierte Perspektive betrachtet wird. Es soll dabei grundsätzlich der gleiche Ansatz wie für imperative Prozessmodelle verfolgt werden, d. h., in vier Schritten (siehe Abschnitt 4.4) wird eine Abbildung zwischen Aktivitäten des Modells festgelegt (Abschnitt 4.4.1): Die durch eine Abbildung bestimmten Elemente werden auf Basis der verschiedenen Perspektiven des Modells miteinander verglichen, aus den verschiedenen Perspektivenähnlichkeiten wird ein globaler Ähnlichkeitswert berechnet werden, der anschließend durch das Anpassen der Abbildung optimiert wird. Da die Abbildung, wie sie in Definition 3.3 vorgegeben ist, lediglich auf der Menge der Aktivitäten definiert ist und für deklarative Prozessmodelle ebenfalls eine Menge an Aktivitäten gegeben ist (siehe Definition 3.2), kann diese Abbildung auch zwischen zwei deklarativen Prozessmodellen oder zwischen einem imperativen und einem deklarativen Prozessmodell festgelegt werden. Auch bei deklarativen Prozessmodellen wird durch die Abbildung eine vollständige, disjunkte Partition der Aktivitätenmenge induziert. Somit ist der erste Schritt des vierstufigen Vorgehens eins zu eins derselbe wie für den Abgleich imperativer Prozessmodelle. Das einzige Modul, das bei einem Abgleich deklarativer Prozessmodelle angepasst bzw. ausgetauscht werden muss, ist das erste der in Abschnitt 4.4.4 genannten: die Wahl der Modellart. Die anderen vier Module sind grundsätzlich dieselben, wobei für die betrachteten Prozessperspektiven für deklarative Modelle geeignete Ähnlichkeitsmaße zu wählen sind. Die Berechnung der Ähnlichkeitswerte der einzelnen Perspektiven, die gemäß der Zielfunktionsdefinition 4.33 miteinander kombiniert werden, wird in den folgenden Abschnitten untersucht.

In Abschnitt 4.5.1 wird auf Ähnlichkeitsabgleiche von deklarativen Prozessmodellen in der Literatur eingegangen. Anschließend erläutert Abschnitt 4.5.2 die Übertragung der Ähnlichkeitsmessmethoden für die funktionale Perspektive und die drei Ressourcenperspektiven. Für die verhaltensorientierte Perspektive sind Übertragungen der imperativen Methoden nicht so einfach möglich. Abschnitt 4.5.3 schlägt mehrere Möglichkeiten vor, das Verhalten von deklarativen Prozessmodellen abzugleichen.

4.5.1 Abgleich von deklarativen Prozessmodellen in der Literatur

Für imperative Prozessmodelle existieren in der Literatur zahlreiche Möglichkeiten, um einen Modellabgleich durchzuführen. Einige davon werden in Kapitel 2 vorgestellt. Für die Ähn-

lichkeitsbestimmung zweier deklarativer Prozessmodelle können der Literatur zwei Ansätze entnommen werden. Zum einen ist es in einigen Fällen möglich, deklarative Prozessmodelle in imperative zu überführen (Prescher et al., 2014), für die dann die in Kapitel 2 genannten Methoden auf Basis von 1:1-Abbildungen unter Einbezug der funktionalen und verhaltensorientierten Perspektive sowie der Struktur Anwendung finden oder die in den Abschnitten 4.1, 4.2 und 4.3 neu eingeführten Methoden, die eine M:N-Abbildung erlauben und alle fünf Perspektiven in den Abgleich einbeziehen. Eine Übersetzung von deklarativen Prozessmodellen in imperative ist jedoch nicht immer möglich bzw. nicht immer ohne einen Verlust der Modellsemantik durchführbar. In vielen deklarativen Sprachen lassen sich Regeln definieren, für die es keine Entsprechung in bestehenden, imperativen Modelliersprachen gibt. Zeising et al. (2014) vergleichen die imperative Modelliersprache BPMN und die deklarative Modelliersprache DPIL bezüglich der Unterstützung verschiedener Modelliermuster innerhalb der Prozessperspektiven und kommen zu dem Ergebnis, dass es sowohl Muster gibt, die von BPMN unterstützt werden, nicht aber von DPIL, und dass es anderherum auch Muster gibt, die DPIL unterstützt, nicht aber BPMN. Das bedeutet, dass sich bei einer Transformation eines Modells in eine andere Sprache bereits Ungenauigkeiten ergeben können, die sich auf den später berechneten Ähnlichkeitswert auswirken und diesen möglicherweise verfälschen.

Wie von Giannakopoulou und Havelund (2001) gezeigt, ist es auch möglich, bestimmte deklarative Prozessmodelle, zum Beispiel solche, die auf LTL basieren, auf endliche Zustandsautomaten (Büchi-Automaten) (Büchi, 1990) abzubilden und dann einen Editierabstand (siehe Abschnitt 2.2.3.10) auf diesen Automaten zu berechnen (Wombacher, 2006). Da in einem Zustandsautomat jedoch Zustände, und nicht Aktivitäten, die Knoten darstellen, ist eine Abwandlung der Abbildung von Aktivitäten auf eine Abbildung von Zuständen erforderlich. Auch ist zu beachten, dass deklarative Modelle, die Variablen benutzen, nicht zwingend in einen endlichen Automaten überführt werden können (Zeising et al., 2014). Es müssten also Ansätze auch für solche Prozessmodelle gefunden werden. Da dieses Vorgehen, wie im Folgenden beschreiben, jedoch weitere Nachteile aufweist, wird es nicht weiter verfolgt.

Eine Übertragung des vierstufigen Abgleichsansatzes auf deklarative Prozessmodelle ist in der Literatur nicht beschrieben, auch nicht der Abgleich der verschiedenen Prozessperspektiven. Implizit werden bei der zuerst genannten Methode, dem Übersetzen von Prozessmodellen, alle die Perspektiven beim Abgleich betrachtet, die im deklarativen Prozessmodell vorhanden sind und in das imperative Modell übernommen werden können. Beim zweitgenannten Ansatz, der Überführung in Zustandsautomaten, ist eigentlich eine Umbenennung der Aktivitätenbeschreibungen in Zustandsbeschreibungen notwendig, auf denen dann labelbasierte Abgleichsmethoden durchgeführt werden können. Gleichzeitig werden, da der Editierabstand der Automaten betrachtet wird, eine strukturelle Ähnlichkeit berechnet. Agenten, Datenobjekte und Werkzeuge werden nicht betrachtet, genauso wenig wie für das Verhalten explizit eine Ähnlichkeit berechnet wird. Die beiden genannten Methoden sind in Abbildung 3.7 bereits mit aufgeführt.

4.5.2 Abgleich der funktionalen Perspektive und der Ressourcenperspektiven

Da die Aktivitätenbeschreibungen über eine Funktion ℓ direkt den Aktivitäten zugeordnet werden (siehe Definition 3.2) und M:N-Abbildungen nur auf Aktivitäten definiert sind, kann eine Berechnung der Labelähnlichkeit (siehe Abschnitt 4.1) direkt von imperativen auf deklarative Prozessmodelle übertragen werden. Ausgenutzt wird hier, dass die Zuweisung von

Aktivitätenbeschreibungen zu Aktivitäten eine Intraaktivitätsrelation (siehe Definition 3.5) darstellt. Die Funktion λ_1 in der Definition für imperative Prozessmodelle und die Funktion ℓ in der Definition für deklarative Prozessmodelle sind gleich aufgebaut, da sie einer Aktivität eine Beschreibung zuordnen. Nicht nur λ_1 sondern auch ℓ erfüllt also die Voraussetzungen für die Berechnung von $BSim$ aus Definition 4.2 und somit kann $BSim$ als Ähnlichkeit der Aktivitätenbeschreibungen auch für deklarative Prozessmodelle angewendet werden.

Für die drei Ressourcenperspektiven gilt Ähnliches wie für die funktionale Perspektive. Die drei Komponenten von λ – λ_2, λ_3 und λ_4 – des imperativen Prozessmodells sind (binäre) Intraaktivitätsrelationen. Sie ordnen einer Aktivität jeweils eine Menge an Agenten, Dokumenten und Werkzeugen zu. Gibt es in einem deklarativen Prozessmodell Regeln in $\mathcal{C}$, die mit binären Intraaktivitätsrelationen der Form $A \times \mathcal{A}$, $A \times \mathcal{D}$ und $A \times \mathcal{S}$ identifiziert werden können, so können mit der gleichen Argumentation wie oben die Methoden $ASim$, $DSim$ und $SSim$ direkt von imperativen auf deklarative Prozessmodelle übertragen werden. Dass es solche Regeln, für die die Identifikation mit den genannten Interaktivitätsrelationen möglich ist, gibt, zeigen die nachfolgenden zwei Beispiele in den Modelliersprachen DECLARE (erweitert um die Datenperspektive) und DPIL.

Beispiel 4.15. In einer Erweiterung von DECLARE, welche Datenobjekte berücksichtigt (Montali et al., 2013), wird eine (nicht atomare) Aktivität als Tupel $(N, \mathcal{P}_s, \mathcal{P}_x, \mathcal{P}_c)$ beschrieben, wobei N die Beschreibung der Aktivität darstellt, mit der sie gleichzeitig identifiziert wird (ℓ ist die Identität), und $\mathcal{P}_i$, der sogenannte Port zum Ereignis i, ein Tupel von folgender Gestalt ist: $\mathcal{P}_i = (E, N, I, D, O)$. Hierbei ist E ein bestimmtes Event (*start* (s), *cancel* (x) oder *complete* (c) einer Aktivität), N ist die Aktivität, I eine Menge von eingehenden Datenobjekten bzw. deren Bezeichnern, D eine Menge an Standardattributen (für dieses Beispiel nicht relevant; z. B. der Ausführende der Aktivität) und O eine Menge von ausgehenden Datenobjekten (deren Bezeichner). In der grafischen Modellierung lassen sich eingehende Datenobjekte nur dem Ereignis *start* einer Aktivität zuordnen, während ausgehende Datenobjekte den Ereignissen *complete* und *cancel* zugeordnet werden. Fasst man die drei Ereignisports einer Aktivität zusammen, so impliziert die Definition der Ports eine Zuweisung von Datenobjekten zu Aktivitäten, nämlich über die Mengen I und O, d. h., $\mathcal{P}_3 \cup \mathcal{P}_5$ einer Aktivität $a \in N$ entspricht also $\lambda_3(\cdot)$ einer Aktivität im imperativen Prozessmodell, was ebenfalls eine Menge an Bezeichnern für Datenobjekte ausgibt. Das Maß $DSim$ lässt sich also auf das um Daten erweiterte DECLARE (Montali et al., 2013) übertragen.

Beispiel 4.16. Die Modelliersprache DPIL ist eine textuelle Sprache, deren Regeln auf Prädikatenlogik erster Stufe (*first order logic*, FOL) basieren. Um den Modellierer bei der Arbeit zu unterstützen, ist es möglich, sogenannte Makros, das sind Kurzausdrücke für Prozessregeln, zu definieren. Wie bereits in Abschnitt 3.2.3 geschrieben, werden von Zeising et al. (2014) einige solcher Makros vorgeschlagen, die die wichtigsten Zusammenhänge, die bei der Modellierung von Prozessen für die verschiedenen Perspektiven benötigt werden, umfassen. Die Makros, die dabei die Ressourcenperspektiven umfassen bzw. deren zugrunde liegenden Formulierungen in FOL, können als Intraaktivitätsrelationen aufgefasst werden, also als Relationen, die einer Aktivität einen Agenten/eine Rolle, ein Datenobjekt oder ein Werkzeug zuordnen. Das Makro *direct(t,i)*, das für den DPIL-Ausdruck *start(of t) implies start(of t by i)* steht, ist eine direkte Zuordnung von Agent i zu Aktivität t. Das Makro *role(t,r)*, das für den DPIL-Ausdruck *start(of t by :i) implies relation(subject i predicate hasRole object r)* steht, besagt, dass Aktivität t von einer Person i ausgeführt werden muss, die Rolle r hat. Es ist also eine

Zuweisung einer Rolle zur einer Aktivität. Dabei sind i bzw. r in $\mathcal{A}$, der Menge der Agenten, aus Definition 3.2 enthalten. Sowohl *direct(·,·)* als auch *role(·,·)* können auf dieselbe binäre Intraaktivitätsrelation wie $\lambda_2(\cdot)$ abstrahiert werden, was eine Übertragung des Ähnlichkeitsmaßes *ASim* von imperativen auf DPIL-Modelle ermöglicht. Eine analoge Beobachtung gilt für die von Zeising et al. (2014) genannten datenbasierten und operationalen Makros.

Da die vier Ähnlichkeitsmaße *BSim*, *ASim*, *DSim* und *SSim* sowohl für imperative als auch für deklarative Prozessmodelle gleichermaßen anwendbar sind, lassen sich, bezüglich dieser vier Perspektiven, auch gemischte Ähnlichkeitsabgleiche, also Abgleiche von imperativen mit deklarativen Prozessmodellen, durchführen.

4.5.3 Abgleich der verhaltensorientierten Perspektive

In deklarativen Prozessmodellen ist, im Unterschied zu imperativen Prozessmodellen, in denen der Kontrollfluss explizit mittels Sequenzflusspfeilen (gerichteten Kanten) und Gateways modelliert ist, der Kontrollfluss implizit über Regeln vorgegeben, genauer: über Regeln, die auf Interaktivitätsrelationen, also auf Relationen, die zwei oder mehr Aktivitäten miteinander verknüpfen, abstrahiert werden können. Bei einem Ähnlichkeitsabgleich der Verhaltensperspektive von deklarativen Modellen müssen also diese Regeln Verwendung finden. Ein Vergleich mit den Methoden, die für imperative Prozessmodelle zur Bestimmung der Verhaltensähnlichkeit zur Verfügung stehen, zeigt, dass viele davon nicht übertragbar sind, da beispielsweise keine expliziten Gatewayinformationen vorhanden sind oder Vorgänger- und Nachfolgerknoten nicht einfach bestimmt werden können. Es werden nun zunächst drei Ansätze genannt, wie ein Abgleich der Verhaltensperspektive durchgeführt werden könnte (Abschnitte 4.5.3.1, 4.5.3.2, 4.5.3.3). Diese Ideen werden von Baumann et al. (2016a) vorgestellt, bringen jedoch einige Probleme mit sich bzw. sind noch unzureichend erforscht, weswegen eine Evaluation dieser Methoden zum momentanen Stand nicht möglich ist. Das Ähnlichkeitsmaß in Abschnitt 4.5.3.1 verwendet Ausführungspfade, das in Abschnitt 4.5.3.2 Abstraktionen von Ausführungspfaden und das in Abschnitt 4.5.3.3 bringt einen neuen Ansatz auf der Ebene der Logik der Modellregeln ins Spiel, der so nur für deklarative Prozessmodelle verwendbar ist. Anschließend wird in Abschnitt 4.5.3.4 eine Übertragung der Methode, die Flussabhängigkeiten verwendet, durchgeführt und auch auf die Eigenschaften der Optionalität und Wiederholbarkeit von Aktivitäten wird kurz eingegangen. Bis auf die Optionalität und Wiederholbarkeit sind alle Methoden nur für 1:1-Abbildungen anwendbar.

4.5.3.1 Ähnlichkeitsabgleich mittels Ausführungspfaden und Konformitätsprüfung

Die erste vorgestellte Möglichkeit verwendet Ausführungspfade von Prozessen bzw. von den zu vergleichenden Prozessmodellen. Sie geht ähnlich vor, wie es bei Process Mining-Verfahren üblich ist, um die Fitness von Logdaten bezüglich der erzeugten Modelle zu bestimmen (Rozinat und van der Aalst, 2006). Es werden simulierte, im besten Fall alle möglichen Ausführungspfade für ein Modell verwendet und dann mit dem zweiten Modell abgeglichen, d. h. auf Ausführbarkeit im zweiten Modell getestet. Da in deklarativen Prozessmodellen jedoch prinzipiell, wie auch für imperative Prozessmodelle (siehe Abschnitt 2.2.4.2), unendlich lange Ausführungspfade möglich sind, kann über Simulation nur eine endliche Menge an möglichen, endlichen Ausführungspfaden erzeugt werden. Dieses Thema der Automated Sequence Generation im Allgemeinen wird beispielsweise von Hallé et al. (2012) behandelt. Ackermann et al.

(2017a) hingegen beschäftigen sich konkret mit der Erzeugung von simulierten Prozesslogs, genauer mit der Simulation von DPIL-Modellen, also von multiperspektivischen, deklarativen Prozessmodellen. Der Simulationsansatz lässt sich jedoch auf beliebige deklarative Prozessmodelle erweitern. Als Charakteristika einer Simulation werden unter anderem genannt, dass zu einer fest vorgegeben Länge *alle* Pfade bis zu dieser Länge erzeugt werden (*exhaustiveness*) und dass auch Gegenbeispiele für Ausführungspfade, also solche Pfade, die in einem Modell nicht möglich sind, generiert werden (*reversibility*). Der von Ackermann et al. (2017a) vorgestellte Simulationsansatz erfüllt das erstgenannte Charakteristikum, somit sind alle anderen Pfade bis zur vorgegebenen Länge, die nicht simuliert werden, automatisch nicht mögliche Ausführungspfade.

Die Repräsentativität des Ergebnisses der Ähnlichkeitsprüfung hängt von der Menge an simulierten Pfaden ab, im Fall der Simulationsmethode von Ackermann et al. (2017a) nur von der angegebenen Länge der möglichen Pfade. Jeder Pfad kann hierbei in eine der vier Kategorien fallen:

- Er ist für beide zu vergleichende Prozessmodelle möglich (K1).
- Er ist für das erste, nicht aber für das zweite Modell möglich (K2).
- Er ist für das zweite, nicht aber für das erste Modell möglich (K3).
- Er ist für keines der beiden Prozessmodelle möglich (K4).

Als Ähnlichkeitsmaß bietet sich der Quotient aus der Anzahl an Pfaden aus K1 zur Anzahl an Pfaden aus K1, K2 und K3 an oder, wenn explizit auch die nicht möglichen Pfade berücksichtigt werden sollen, der Quotient $^{(|K1|+|K4|)}/_{(|K1|+|K2|+|K3|+|K4|)}$. Die letztgenannte Methode führt zwar einen Abgleich über alle Pfade durch, ist so gesehen vollständig, jedoch erhalten auch diejenigen Prozessmodelle eine positive Ähnlichkeit, die keine gemeinsamen Pfade haben, sondern lediglich bestimmte Pfade gleichermaßen verbieten. Die erste Methode ist somit zu bevorzugen und wird auch im nachfolgenden Beispiel, das dem Ähnlichkeitsmaß den Namen $VSim^{ep}$ gibt, verwendet. Sie entspricht der Berechnung des Jaccard-Koeffizienten, wenn die jeweils möglichen Ausführungspfade als Mengen an Ausführungspfaden aufgefasst werden.

Sind alle diese Zahlen, also $|K_1|, |K_2|, |K_3|$ und, falls berücksichtigt, $|K_4|$ endlich, was sie bei oben genanntem Simulationsverfahren mit vorgegebener maximaler Pfadlänge sind, kann das Maß einfach berechnet werden. Sind die Anzahlen jedoch unendlich, existiert lediglich ein Grenzwert. Es kann dabei vorkommen, dass dieser Grenzwert immer aus $\{0, 1\}$ ist.

Beispiel 4.17. Gegeben seien die beiden Prozessmodelle in Tabelle 4.8, die an die DPIL-Notation angelehnt sind, jedoch keine DPIL-Modelle sind. In beiden Modellen ist nur die funktionale und die verhaltensorientierte Perspektive modelliert. Modell S_1 beinhaltet die drei Aktivitäten A, B und C und fünf Regeln, die Folgendes aussagen:

- c_1: A muss mindestens einmal und darf maximal zweimal ausgeführt werden.
- c_2: B muss genau einmal ausgeführt werden.
- c_3: C darf höchstens einmal ausgeführt werden.
- c_4: Wenn B ausgeführt werden soll, muss A vorher bereits beendet sein.

- c_5: Wenn C beendet wird, muss später auch noch B beendet werden.

Modell S_2 beinhaltet die drei Aktivitäten A', B' und C', wobei die (1:1-)Abbildung $M : A_1 \to A_2$ aus $M = \{(A, A'), (B, B'), (C, C')\}$ besteht, und fünf Regeln, deren Bedeutung wie folgt ist:

- c'_1: A' muss mindestens einmal und darf maximal zweimal ausgeführt werden.
- c'_2: B' muss genau einmal ausgeführt werden.
- c'_3: C' muss genau einmal ausgeführt werden.
- c'_4: Wird A' ausgeführt, muss später auch noch C' ausgeführt werden.
- c'_5: Wird B' begonnen, muss C' bereits abgeschlossen sein und sobald C' abgeschlossen wird, muss auch B' später noch beendet werden.

Für beide Modelle gilt, dass Ausführungspfade eine maximale Länge von 4 haben dürfen. Für S_1 ergeben sich daraus folgende erlaubte Ausführungspfade, die zur Menge ν_1 zusammengefasst werden, wobei nicht zwischen *start* und *complete* der Aktivitäten unterschieden wird: $\nu_1 = \{AB,\ AAB,\ ABA,\ CAB,\ ACB,\ AACB,\ ACAB,\ CAAB,\ ACBA,\ CABA\}$. Für S_2 ergeben sich die Ausführungspfade $\nu_2 = \{A'C'B',\ A'A'C'B'\}$. Werden in ν_2 die Aktivitäten mit ihren Urbildern aus A_1 identifiziert, so ist die Ähnlichkeit von S_1 und S_2

$$VSim^{ep}(S_1, S_2) = \frac{|\nu_1 \cap \nu_2|}{|\nu_1 \cup \nu_2|} = \frac{2}{10} = 0,2.$$

Dieser Wert scheint nicht besonders hoch[3], jedoch gilt im Beispiel, dass $\nu_2 \subseteq \nu_1$, was bedeutet, dass S_2 eine Einschränkung von S_1 darstellt. Diese Information kann zusätzlich zum eigentlichen Ähnlichkeitswert nützlich sein.

Tabelle 4.8: Beispiel zweier deklarativer Prozessmodelle mit Regeln, die die Häufigkeit der Ausführung und eine mögliche Reihenfolge der Aktivitäten betreffen.

	Aktivitäten $\mathcal{A}$	Regeln $\mathcal{C}$
S_1	A, B, C	c_1: $1 \leq count(start\_of(A)) \leq 2$
		c_2: $count(start\_of(B)) = 1$
		c_3: $count(start\_of(C)) \leq 1$
		c_4: $start\_of(B) \to_{t>} complete\_of(A)$
		c_5: $complete\_of(C) \to_{t<} complete\_of(B)$
S_2	A', B', C'	c'_1: $1 \leq count(start\_of(A')) \leq 2$
		c'_2: $count(start\_of(B')) = 1$
		c'_3: $count(start\_of(C')) = 1$
		c'_4: $complete\_of(A') \to_{t<} complete\_of(C')$
		c'_5: $start\_of(B') \to_{t>} complete\_of(C') \wedge$
		$complete\_(C') \to_{t<} complete\_of(B')$

[3]Dies stellt eine rein qualitative Einschätzung dar. Eine detailliertere Evaluation ist zum gegenwärtigen Zeitpunkt noch nicht möglich. Zum Vergleich wird in Abschnitt 4.5.3.2 für dieselben Modelle mit einer anderen Methode ein Ähnlichkeitswert berechnet.

Da eine Simulation auch für imperative Prozessmodelle (siehe Abschnitt 2.2.4.2) möglich ist, können mit dieser Methode auch imperative und deklarative Prozessmodelle miteinander verglichen werden. Korrespondenzen zwischen konkreten Ausführenden, Datenobjekten und Services müssen, neben der Abbildung, die Aktivitäten aufeinander abbildet, gegeben sein.

4.5.3.2 Ähnlichkeitsabgleich mittels Verhaltensmustern und Wahrheitstabellen

Die zweite Möglichkeit, Verhalten von deklarativen Prozessmodellen abzugleichen, ist eine Approximation der Methode aus Abschnitt 4.5.3.1 und stammt ebenfalls aus der Arbeit von Baumann et al. (2016a). Es werden zunächst für die abgebildeten Aktivitäten vorher festgelegte Verhaltensmuster bestimmt, die eine bestimmte Anzahl z an Aktivitäten umfassen und eine bestimmte (maximale) Länge l haben, wobei $z \leq l$ gewählt wird, damit auch mindestens ein Muster, in dem alle z verschiedenen Aktivitäten gleichzeitig auftauchen, berücksichtigt wird. Für jeweils bis zu z verschiedene Aktivitäten werden dann alle Permutationen bis zur Länge l gebildet, wobei Wiederholungen der Aktivitäten erlaubt sind (und notwendig sind, wenn $z < l$). Diese Permutationen stellen Ausschnitte aus potentiellen Ausführungspfaden dar. Bei bis zu 3-stelligen Mustern ($l = 3$) für die Aktivitäten A und B ($z = 2$) ergeben sich beispielsweise folgende Muster, wobei nur positive Zusammenhänge berücksichtigt werden: $-, A, B, AA, AB, \ldots, BBB$. Es wird dann mit Hilfe der Regelmengen beider Prozessmodelle jedem Muster ein Wahrheitswert zugewiesen, d. h., es wird geprüft, ob das Muster im jeweiligen Modell so ausführbar ist. Ein einzelnes A würde in diesem Fall bedeuten, dass A irgendwann einmal ausgeführt wird und davor und danach weder A noch B ausgeführt werden (andere, an diesem Muster nicht beteiligte Aktivitäten, können aber durchaus ausgeführt werden). Es wird hier zunächst nicht unterschieden, ob die Ausführungen direkt oder irgendwann nacheinander erfolgen müssen.

Sind die Wahrheitswerte für die Muster in beiden Modellen bestimmt, werden sie abgeglichen: Ist ein Muster in beiden Modellen erlaubt oder in beiden Modellen nicht erlaubt, wird ihm der Wert 1 zugewiesen; ist ein Muster in einem Modell erlaubt, im andern nicht, wird ihm der Wert 0 zugewiesen. Diese Werte werden addiert und durch die Anzahl der Muster geteilt, was einen Mittelwert im Intervall $[0, 1]$ liefert. Erlauben beide Modelle die gleichen Muster, ist der Wert 1. Je weniger Muster übereinstimmen, desto mehr tendiert der Mittelwert zu 0. Im Grunde entspricht diese Methode einer Art Vergleich von Fragmenten von Ausführungspfaden (siehe auch Abschnitt 2.2.4.3, wo für imperative Prozessmodelle etwas Ähnliches vorgeschlagen wird). Anders als in Abschnitt 4.5.3.1, bei dem die Pfade, die in beiden Prozessmodellen nicht möglich sind, bei der Ähnlichkeitsberechnung letztendlich nicht berücksichtigt werden, werden diese hier mit einberechnet, da die Menge der Muster, die überprüft wird, im Allgemeinen nicht vollständig ist. Aus der Tatsache, dass es keine gemeinsamen Ausführungspfade gibt, kann nicht darauf geschlossen werden, dass es keine gemeinsamen Muster gibt. Das nachfolgende Beispiel berechnet die Ähnlichkeit der Prozessmodelle aus Tabelle 4.8 unter Verwendung von maximal zweistelligen Mustern über je zwei Aktivitäten ($z = 2, l = 2$). Dieses Ähnlichkeitsmaß sei mit $VSim^{pat}$ bezeichnet. In den Zellen der Tabelle vermerkt ist der Wahrheitswert des jeweilien Musters bezogen auf S_1 bzw. S_2.

Beispiel 4.18. Die Tabelle 4.9 zeigt die Erfüllbarkeit der Muster für die beiden Prozessmodelle aus Tabelle 4.8, wobei $z = 2$ und $l = 2$ gesetzt ist. Die zugehörige Wahrheitstabelle ist in Tabelle 4.10 dargestellt. Das Ähnlichkeitsmaß ist damit

$$VSim^{pat}(S_1, S_2) = \frac{17}{21} \approx 0,81.$$

Dieser Wert ist viel höher als der, der mittels der Ausführungspfade errechnet wird (Abschnitt 4.5.3.1). Dies liegt zum einen daran, dass für dieses Beispiel nur die kürzestmöglichen Muster verwendet wurden, zum anderen daran, dass beide Prozessmodelle überhaupt nur relativ wenige Ausführungsmöglichkeiten erlauben und deshalb der Anteil der Muster, die in beiden Modellen nicht erlaubt sind, sehr hoch ist. Dies ist in Tabelle 4.9 gut zu erkennen. Die Ausführungspfade, die in beiden Modellen nicht erlaubt sind, sind bei der Berechnung von $VSim^{ep}$ nicht berücksichtigt. Würden diese hier weggelassen werden, wäre der Wert bei $\frac{3}{7}$, was ungefähr $0,43$ ist. Anhand der Erfüllbarkeitstabelle lässt sich aber auch über diese Methode erkennen, dass dort, wo Diskrepanzen auftreten, S_1 das Modell ist, das mehr verschiedene Ausführungen zulässt, also das allgemeinere Modell ist.

Tabelle 4.9: Erfüllbarkeitstabelle für die zwei Beispielprozesse in Tabelle 4.8 mit maximal zweistelligen Mustern für je zwei Aktivitäten.

S_1, S_2	$-$	x	y	xx	xy	yx	yy
$x = A, y = B$	-,-	-,-	-,-	-,-	+,+	-,-	-,-
$x = A, y = C$	-,-	+,-	-,-	+,-	+,+	+,-	-,-
$x = B, y = C$	-,-	+,-	-,-	-,-	-,-	+,+	-,-

Tabelle 4.10: Wahrheitstabelle für die zwei Beispielprozesse in Tabelle 4.8 zur Erfüllbarkeitstabelle 4.9 mit maximal zweistelligen Mustern für je zwei Aktivitäten.

S_1, S_2	$-$	x	y	xx	xy	yx	yy
$x = A, y = B$	1	1	1	1	1	1	1
$x = A, y = C$	1	0	1	0	1	0	1
$x = B, y = C$	1	0	1	1	1	1	1

Eine wichtige Frage, die sich für diese Möglichkeit noch stellt, ist die, welche Verhaltensmuster für den Abgleich verwendet werden sollen, um ein möglichst objektives und verallgemeinerbares Ergebnis zu erhalten, und wie diese Muster überprüft werden können. Die Methode bietet nur dann einen Vorteil gegenüber der aus Abschnitt 4.5.3.1, wenn die Überprüfung schneller und einfacher als die Erzeugung/Simulation von Ausführungspfaden abläuft. Ein Vorteil dieser Methode ist aber, dass Ausführungen mit beliebigen Schleifen ohne Einschränkung zugelassen sind. Grundsätzlich ist diese Methode, mittels Verhaltensmustern einen Abgleich durchzuführen, auch für imperative Prozessmodelle anwendbar, die Methode kann also auch für einen gemischten Abgleich verwendet werden.

Nach den beiden auf Pfaden basierenden Abgleichsmethoden für deklarative Prozessmodelle, wird in Abschnitt 4.5.3.3 ein komplett anderer Ansatz vorgeschlagen. Dieser arbeitet direkt auf der Menge $\mathcal{C}$ der Regeln, die das Verhalten und die Ressourcenzuordnungen der deklarativen Prozessmodelle definieren. Er ist als alternative Möglichkeit zu verstehen, die an dieser Stelle angesprochen werden soll, auf die jedoch im weiteren Verlauf nicht näher eingegangen wird.

4.5.3.3 Ähnlichkeitsprüfung mittels Prädikatenlogik und Regelähnlichkeit

Eine weitere Möglichkeit, wie die Verhaltensähnlichkeit von zwei deklarativen Prozessmodellen bestimmt werden kann, ist die, die Regeln der Prozessmodelle direkt auf Ebene der zugrunde liegenden Logik miteinander zu vergleichen.[4] Es stellt sich die Frage, ob aus der Logik bekannte Erfüllbarkeits- bzw. Entscheidbarkeitskriterien für eine Ähnlichkeitsbestimmung benutzt werden können, wie zum Beispiel in Bisson (1992) oder Sebag und Schoenauer (1994) für statistische Lernverfahren durchgeführt. Nach einer ersten Einschätzung ergibt sich für die Regelmengen deklarativer Prozessmodelle, zumindest dann, wenn sie Aussagen über komplexere Zusammenhänge erlauben, dass sie unentscheidbar werden, weswegen dieser Ansatz bislang auch nicht weiterverfolgt wurde und auch in der vorliegenden Arbeit nicht vertieft wird. Der Vergleich kann (gegeben entscheidbare Regelmengen) so erfolgen: Kann eine Regel aus S_1 aus den Regeln von S_2 abgeleitet werden, wird sie mit 1 bewertet, sonst mit 0, und andersherum. Dann werden diese Werte aufsummiert und durch die Anzahl der Regeln geteilt. Diese Anzahl der Regeln stellt hierbei aber auch ein Problem dar, da Regelmengen, die zu einem gleichen Verhalten führen, nicht identisch, also vor allem nicht gleich mächtig, sein müssen. Das heißt, sowohl Dividend als auch Divisor in diesem Ähnlichkeitsmaß sind nicht eindeutig bestimmt. Außerdem wäre eine solche Methode nicht auf imperative Prozessmodelle übertragbar, außer diese können in derselben Logik ausgedrückt werden, was einer Modelltransformation gleichkommt. Wenn die Zuordnung von Ressourcen zu Aktivitäten mittels der Regeln erfolgt, könnten die Ressourcenperspektiven hier allerdings gleich mit berücksichtigt werden. Auch perspektivenübergreifende Regeln bzw. Regeln, die erst zur Laufzeit vollständig ausgewertet werden können, könnten so möglicherweise Berücksichtigung finden, was mit den anderen Methoden nicht in vollem Umfang möglich ist.

4.5.3.4 Ähnlichkeit mittels Flussabhängigkeiten in deklarativen Prozessmodellen

Die Abgleichsmethode mittels Flussabhängigkeiten, die in Abschnitt 4.3.6 für imperative Prozessmodelle eingeführt wird, kann auf deklarative Prozessmodelle übertragen werden, sofern aus den Regeln des deklarativen Modells die genannten Flussabhängigkeiten abgeleitet werden können. Die Flussabhängigkeiten, auf denen das Ähnlichkeitsmaß definiert ist, sind nichts anderes als Interaktivitätsrelationen, sodass die Methodenübertragbarkeit gemäß Abschnitt 3.5.3 möglich ist. Da die Regeln von deklarativen Prozessmodellen jedoch keiner einheitlichen Form genügen, kann eine Ableitung in dieser Arbeit nur beispielhaft erfolgen. Hierzu sind folgende, wieder an DPIL angelehnte Regelvorlagen vorgegeben, aus denen ein deklaratives Prozessmodell gebildet werden kann. Die Regeln unterteilen sich in Kontrollflussregeln und Datenregeln und sind in Tabelle 4.11 gegeben.

Die insgesamt 24 Regelvorlagen aus Tabelle 4.11 sind so zu lesen, dass jedes s/c_of entweder als $start\_of$ oder $complete\_of$ gelesen werden kann. Die Implikationen können rein existentiell sein (ohne zeitliche Bedingung) oder mit einer zeitlichen Bedingung versehen. Hierbei bedeutet $\rightarrow_{t<}$, dass die Folgerung zeitlich nach der Bedingung geschehen muss, $\rightarrow_{t>}$ hingegen, dass die Folgerung zeitlich vor der Bedingung passiert sein muss. Eine Folgerung kann auch das Nichtauftreten eines Ereignisses sein. Jede Implikation $A \rightarrow B$ kann auch als $\neg(A \wedge \neg B)$ oder $\neg A \vee B$ geschrieben werden. Außerdem gilt $A \rightarrow B = \neg B \rightarrow \neg A$, was es ermöglicht, alle Regeln mit positiver Bedingung zu formulieren. Dies gilt auch bei zeitlichen Implikationen, da $\rightarrow_{t\leqslant}$

[4] Genau genommen beziehen die Regeln auch die drei Ressourcenperspektiven mit ein, es könnten also alle Perspektiven bis auf die funktionale direkt über die Regelmenge verglichen werden.

Tabelle 4.11: Liste mit 24 Regelvorlagen, wobei A und B Aktivitäten bezeichnen und δ und ε Datenobjekte; s/c steht für *start* bzw. *complete*; $t<$ steht für eine zeitliche Nachbedingung, d. h., $A \rightarrow_{t<} B$ bedeutet, dass B zeitlich nach A gilt, $t>$ für eine zeitliche Vorbedingung.

Kontrollflussregeln	Datenregeln
$s/c\_of(A) \rightarrow_{(t\leqslant)} (not)s/c\_of(B)$	$s/c\_of(A) \rightarrow_{(t\leqslant)} (not)write\_of(\delta)$
	$write\_of(\delta) \rightarrow_{(t\leqslant)} (not)s/c\_of(B)$
	$write\_of(\delta) \rightarrow_{(t\leqslant)} (not)write\_of(\varepsilon)$

lediglich eine Kurzschreibweise ist. Der Ausdruck $start\_of(A) \rightarrow_{t<} not\ complete\_of(B)$ kann auch als $start\_of(A)\ at\ t \rightarrow (not\ complete\_of(B)\ at\ s) \wedge t < s$ geschrieben werden. Die Unterscheidung der verschiedenen Ereignisse einer Aktivität (hier: *start* und *complete*) wird üblicherweise bei deklarativen Prozessmodellen gemacht, jedoch wirkt sich die Unterscheidung, wie nachfolgend erkennbar sein wird, hier nicht auf die Ähnlichkeitsbestimmung aus. An dieser Stelle besteht also schon ein erster Abstraktionsschritt bei der Betrachtung der zu vergleichenden Prozessmodelle.

Beispiel 4.19. Tabelle 4.12 zeigt ein deklaratives Prozessmodell, das mit Hilfe der Regelvorlagen aus Tabelle 4.11 formuliert ist. Es besteht aus insgesamt acht Regeln, zu denen jeweils eine kurze Erklärung der Bedeutung gegeben ist. Regeln 1, 2, 3 und 7 betreffen rein den Kontrollfluss, Regeln 4, 5, 6 und 8 beziehen Datenobjekte mit ein. Aus ihnen kann, wie später gezeigt, ein Datenfluss abgeleitet werden.

Für sämtliche Regeln aus Tabelle 4.11 werden nun die kausalen und ordnenden Flussabhängigkeiten aus Definition 4.25 hergeleitet. Das heißt, es wird eine Übersetzung jeder Regel in Flussabhängigkeiten angegeben. Da von den drei Ressourcenperspektiven nur die datenorientierte Perspektive in diesem Abschnitt relevant ist, werden die organisatorische und die operationale Perspektive der Übersichtlichkeit wegen weggelassen. Auch die Aktivitätenbeschreibungen spielen keine Rolle, weswegen die Label und die Zuordnungsfunktion der Label zu Aktivitäten ebenfalls nicht dargestellt wird. Die Übersetzung der reinen Kontrollflussregeln ist in Definition 4.41 angegeben:

Definition 4.41 (Transformation von Kontrollflussregeln in Flussabhängigkeiten)**.** Es sei $S = (A, \mathcal{D}, \mathcal{C})$ ein deklaratives Prozessmodell mit Regelmenge $\mathcal{C}$ wie in Tabelle 4.11. Aus den Kontrollflussregeln werden folgende Flussabhängigkeiten abgeleitet:

- $s/c\_of(A) \rightarrow s/c\_of(B)$: $=> (A,B)$ und $\leftrightarrow (A,B)$ und $\leftrightarrow (B,A)$
- $s/c\_of(A) \rightarrow_{t>} s/c\_of(B)$: $=> (A,B)$ und $\leftrightarrow (A,B)$ und $\twoheadleftarrow (B,A)$
- $s/c\_of(A) \rightarrow_{t<} s/c\_of(B)$: $=> (A,B)$ und $\twoheadrightarrow (A,B)$ und $\leftrightarrow (B,A)$
- $s/c\_of(A) \rightarrow not\ s/c\_of(B)$: $>< (A,B)$ und $-(A,B)$ und $-(B,A)$
- $s/c\_of(A) \rightarrow_{t>} not\ s/c\_of(B)$: $\sim (A,B)$ und $\leftrightarrow (A,B)$ und $-(B,A)$
- $s/c\_of(A) \rightarrow_{t<} not\ s/c\_of(B)$: $\sim (A,B)$ und $-(A,B)$ und $\leftrightarrow (B,A)$

Eine symmetrische Existenzabhängigkeit $==$ wird aus zwei asymmetrischen Existenzabhängigkeiten gefolgert: $== (A,B)$ folgt aus $=> (A,B)$ und $=> (B,A)$. Beeinflussen mehrere Regeln dieselben zwei Aktivitäten, so dominieren $\twoheadrightarrow$ bzw. $-$ das standardmäßige $\rightarrow$; Gleiches gilt für $=>$ bzw. $><$, die $\sim$ dominieren.

Tabelle 4.12: Beispiel eines deklarativen Prozessmodells mit acht Regeln.

Nr.	Regel	Erklärung
1	$start\_of(B) \rightarrow_{t>} complete\_of(A)$	A muss abgeschlossen sein, bevor mit B begonnen werden kann.
2	$complete\_of(B) \rightarrow not\ complete\_of(C)$	B kann nur beendet werden, wenn C noch nicht abgeschlossen ist; sobald B abgeschlossen ist, kann C nicht mehr beendet werden.
3	$complete\_of(C) \rightarrow not\ complete\_of(B)$	C kann nur beendet werden, wenn B noch nicht abgeschlossen ist; sobald C abgeschlossen ist, kann B nicht mehr beendet werden.
4	$start\_of(F) \rightarrow_{t>} write\_of(\delta)$	F kann nur gestartet werden, wenn δ bereits verfügbar ist.
5	$complete\_of(E) \rightarrow write\_of(\delta)$	Der Abschluss von E verlangt, dass δ geschrieben wird.
6	$start\_of(G) \rightarrow write\_of(\delta)$	Das Starten von G verlangt, dass δ geschrieben wird.
7	$complete\_of(H) \rightarrow complete\_of(G)$	Immer wenn H beendet wird, muss auch G irgendwann beendet sein.
8	$write\_of(\delta) \rightarrow not\ complete\_of(D)$	Sobald δ erzeugt wurde, kann D nicht mehr beendet werden. Daraus folgt auch unmittelbar, dass sobald D beendet wurde, δ nicht mehr erzeugt werden kann.

Konflikte können bei einem widerspruchsfreien Modell nicht auftreten, d. h., es kann nicht vorkommen, dass zum Beispiel sowohl $=> (A, B)$ als auch $>< (A, B)$ aus den Regeln abgeleitet wird.[5] Unter anderem an der Tatsache, dass bei den Flussabhängigkeiten nicht danach unterschieden wird, ob eine Aktivität über ihr Start- oder Beendigungsereignis an einer Regel beteiligt ist, wird erkennbar, dass es sich bei der Bestimmung der Flussabhängigkeiten um eine Abstraktion des Modells bzw. seines Verhaltens handelt.

Definition 4.42 (Transformation von Datenregeln in Flussabhängigkeiten)**.** Es sei $S = (A, \mathcal{D}, \mathcal{C})$ ein deklaratives Prozessmodell mit Regelmenge $\mathcal{C}$ wie in Tabelle 4.11. Die Datenregeln werden mittels Transitivität des zeitlichen Aspekts und Anwendung des Kalküls des natürlichen Schließens auf bekannte Kontrollflussregeln zurückgeführt. Datenflussabhängigkeiten werden dann auf dieselbe Weise wie Kontrollflussabhängigkeiten hergeleitet und mit einem hochgestellten d versehen.

Für eine Anwendung von Definition 4.42 betrachte die Regeln 4 und 8 des Beispielmodells aus Tabelle 4.12. Es sei angenommen dass Aktivität F zum Zeitpunkt t' ausgeführt wird $(F_{t'})$. Mit dem Kalkül des natürlichen Schließens ergibt sich durch spezielle Beseitigungs-

[5] Bei perspektivenübergreifenden Regeln könnte solch ein Fall durchaus auftreten, z. B.: „Wenn $\delta < x$, dann muss, wenn A ausgeführt wird, auch B gemacht werden; wenn $\delta \geq x$, dann darf B nicht gemacht werden, falls A ausgeführt wird.“ Solch eine Regel kann aktuell über die Flussabhängigkeiten nicht abgebildet werden und wäre z. B. mit $\sim (A, B)$ aufzulösen.

und Einfügeregeln (mit B und E kenntlich gemacht), nämlich durch Anwendung von Modus ponens ($\rightarrow B$), Beseitigung der Konjunktion ($\wedge B$) und Konditionaleinführung ($\rightarrow E$), folgende Herleitung: Aus $F_{t'} \rightarrow \delta_{t''} \wedge t'' < t'$ sowie $\delta_{t''} \rightarrow \neg D_{t'''}$ folgt $F_{t'} \rightarrow \neg D_{t'''}$. Die Darstellung als Beweisbaum, der von oben nach unten zu lesen ist,[6] lautet:

$$\dfrac{\dfrac{\dfrac{(F_{t'}) \qquad F_{t'} \rightarrow \delta_{t''} \wedge t'' < t'}{\delta_{t''} \wedge t'' < t'} \rightarrow B}{\delta_{t''}} \wedge B \qquad \delta_{t''} \rightarrow \neg D_{t'''}}{\dfrac{\neg D_{t'''}}{F_{t'} \rightarrow \neg D_{t'''}} \rightarrow E} \rightarrow B$$

Da t' und t''' unabhängig voneinader sind, ist die hergeleitete Regel also $start\_of(F) \rightarrow not\ complete\_of(D)$. Daraus ergibt sich, unter Anwendung von Definition 4.41 die datenbasierten Flussabhängigkeiten ${}^{d}><(F,D)$, ${}^{d}-(F,D)$ und ${}^{d}-(D,F)$.

Wegen der Transitivitätseigenschaften verschiedener Flussabhängigkeiten lassen sich nicht nur für die direkt in den Regeln spezifizierten Aktivitäten Abhängigkeiten herleiten, sondern darüber hinaus möglicherweise auch für weitere Aktivitätenpaare (sog. *hidden dependencies* in deklarativen Prozessmodellen (De Smedt et al., 2016)). So kann iterativ die Abhängigkeitsmatrix mit Flussabhängigkeiten gefüllt werden. Die deklarative Herangehensweise an die Modellierung ist die, dass grundsätzlich alles, was nicht verboten oder irgendwie eingeschränkt ist, erlaubt ist. Gemäß dieses Ansatzes ist die Initialbelegung der Flussabhängigkeiten für ein Aktivitätentupel $\sim(\cdot,\cdot)$ für die kausale Abhängigkeit und $\leftrightarrow(\cdot,\cdot)$ für die ordnende Abhängigkeit. Diese Initialbelegungen können von strengeren Flussabhängigkeiten überschrieben werden. Eine Auflistung aller Transitivitäten innerhalb der Flussabhängigkeiten ist nachfolgend gegeben. Hierbei sind nur die Transitivitäten aufgeführt, die die ursprünglichen Abhängigkeiten verändern. Aus bestimmten Kombinationen aus kausalen und ordnenden Abhängigkeiten zwischen je zwei Aktivitäten (A,B) und (B,C) lassen sich Abhängigkeiten für (A,C) herleiten. Nur wenn beide Ursprungsabhängigkeiten datenbezogen sind, so ist auch die abgeleitete Abhängigkeit datenbezogen.

- Es existieren die positiven Abhängigkeiten

$$=>(a,b) \wedge =>(b,c) \Rightarrow =>(a,c),$$
$$\twoheadrightarrow(a,b) \wedge \twoheadrightarrow(b,c) \Rightarrow \twoheadrightarrow(a,c)$$
$$\twoheadleftarrow(a,b) \wedge \twoheadleftarrow(b,c) \Rightarrow \twoheadleftarrow(a,c)$$

- Es existieren die negativen Abhängigkeiten

$$=>(a,b) \wedge ><(b,c) \Rightarrow ><(a,c), -(a,c), -(c,a)$$
$$\twoheadrightarrow(a,b) \wedge -(c,b) \Rightarrow -(c,a)$$
$$\twoheadleftarrow(a,b) \wedge -(a,c)) \Rightarrow -(b,c)$$

Die direkt herleitbaren Abhängigkeiten des Beispielmodells aus Tabelle 4.12 sind die, die in Abhängigkeitstabelle 4.13 eingetragen sind. Die Standardbelegung ist der Übersichtlichkeit halber hier nicht sichtbar. Über Transitivitäten lassen sich hieraus die Abhängigkeiten wie in

[6] Die Formeln, die keine Vorgänger haben und nicht eingeklammert sind, sind die Voraussetzungen, der Ausdruck ganz unten ist das Gefolgerte

Tabelle 4.13: Direkt ableitbare Flussabhängigkeiten aus dem deklarativen Prozessmodell aus Tabelle 4.12.

	A	B	C	D	E	F	G	H
A		$<=$ $\nleftrightarrow$						
B	$=>$ $\leftrightarrow$		$><$ $-$					
C		$><$ $-$						
D					${}^{d}><$ ${}_{d}-$	${}^{d}><$ ${}_{d}-$	${}^{d}><$ ${}_{d}-$	
E				${}^{d}><$ ${}_{d}-$				
F				${}^{d}><$ ${}_{d}-$				
G				${}^{d}><$ ${}_{d}-$				$<=$ $\leftrightarrow$
H							$=>$ $\leftrightarrow$	

Tabelle 4.14 zu sehen zusätzlich eintragen (betrifft (D, H) bzw. (H, D)). Vollständig mit den unbestimmten Abhängigkeiten gefüllt ergibt sich die Abhängigkeitsmatrix wie in Tabelle 4.15 dargestellt.

Für das imperative Prozessmodell aus Abbildung 4.12 mit seiner Abhängigkeitsmatrix aus Tabelle 4.4 und das deklarative Prozessmodell aus Tabelle 4.12 mit seiner Abhängigkeitsmatrix aus Tabelle 4.15 ist die Abgleichsmatrix in Tabelle 4.16 dargestellt. Es gibt keine Widersprüche bei diesem Vergleich. Was jedoch auffällt, ist die Subsumptionseigenschaft des einen Modells. In diesem Fall ist im deklarativen Modell all das möglich, was auch im imperativen möglich ist, aber zusätzlich noch mehr. Das heißt, das deklarative Modell ist eine Erweiterung des imperativen Modells bzw. das imperative stellt eine echte Einschränkung des deklarativen dar. Gemäß Definition 4.32 kann zusätzlich ein Ähnlichkeitsmaß ausgerechnet werden. Wie beim Vergleich zweier imperativer Prozessmodelle, ist auch bei einem Abgleich von einem imperativen und einem deklarativen Modell das Einbeziehen der Datenabhängigkeiten nicht ratsam. Werden jedoch zwei deklarative Modelle abgeglichen, kann diese Information berücksichtigt werden. Wie im deklarativen Beispielmodell aus Tabelle 4.12 gesehen, ist es in deklarativen Modellen durchaus der Fall, dass reine Datenabhängigkeiten bestehen.

Wie am Ende von Abschnitt 4.3.6 für imperative Modelle vorgeschlagen, können auch für deklarative Modelle in der Abgleichs- bzw. Wertungsmatrix auf der Diagonalen Informationen über Optionalität und Wiederholbarkeit der einzelnen Aktivitäten hinzugefügt werden, da diese aus den Flussabhängigkeiten im Allgemeinen nicht ableitbar sind und somit in die Berechnung von $VSim^{fdep}(\cdot, \cdot)$ nicht einfließen. Wie die Optionalität und die Wiederholbarkeit einer Aktivität bestimmt wird, hängt von der zugrunde liegenden Modelliersprache ab. Baumann et al. (2016b) beschreiben eine Methode, wie zu jedem Zeitpunkt der Ausführung eines deklarativen Prozessmodells in DECLARE bzw. ConDec auf Basis der Regeln und der

Tabelle 4.14: Direkt ableitbare und transitiv abgeleitete Flussabhängigkeiten aus dem deklarativen Prozessmodell aus Tabelle 4.12.

	A	B	C	D	E	F	G	H
A		$<=$ $\nleftrightarrow$						
B	$=>$ $\leftrightarrow$		$><$ $-$					
C		$><$ $-$						
D					$\overset{d}{>}<$ $\underset{d}{-}$	$\overset{d}{>}<$ $\underset{d}{-}$	$\overset{d}{>}<$ $\underset{d}{-}$	$><$ $-$
E				$\overset{d}{>}<$ $\underset{d}{-}$				
F				$\overset{d}{>}<$ $\underset{d}{-}$				
G				$\overset{d}{>}<$ $\underset{d}{-}$				$<=$ $\leftrightarrow$
H				$><$ $-$			$=>$ $\leftrightarrow$	

bisherigen Ausführungshistorie bestimmt werden kann, wie oft eine Aktivität noch mindestens ausgeführt werden muss, bevor der Prozessdurchlauf erfolgreich beendet ist, und wie oft eine Aktivität noch maximal ausgeführt werden kann. Diese Werte existieren auch für die leere Ausführungshistorie, also für den Anfang einer Prozessausführung, bevor die erste Aktivität ausgeführt wurde. Ein Minimalwert von 0, der bedeutet, dass eine Aktivität gar nicht ausgeführt werden muss, um den Prozessdurchlauf erfolgreich zu beenden, entspricht also einer optionalen Aktivität. Ein Maximalwert von > 1 bedeutet, dass eine Aktivität mehr als einmal ausgeführt werden kann, sie also wiederholbar ist. Für Modelle in DECLARE bzw. ConDec kann somit die Optionalität bzw. Wiederholbarkeit in die Ähnlichkeitsberechnung mittels Flussabhängigkeiten aufgenommen werden. Analog zu $VSim^{\rho}$ und $VSim^{o}$ aus den Abschnitten 4.3.2 und 4.3.3 können aber auch unter M:N-Abbildungen eine Optionalitätsähnlichkeit und eine Wiederholbarkeitsähnlichkeit für deklarative Prozessmodelle berechnet werden. Eine Positionsähnlichkeit ($VSim^{\pi}$) ist, da die Definition der relativen Position einer Aktivität auf deklarative Prozessmodelle nicht übertragbar ist, nicht bestimmbar.

Neben der Einschränkung auf 1:1-Abbildung beim Ähnlichkeitsansatz mittels Flussabhängigkeiten, existiert auch die Einschränkung, dass instanzbasierte (Daten-)Regeln mit diesem abstrahierenden Ansatz nicht vollständig berücksichtigt werden können. Montali et al. (2013) formulieren beispielsweise eine datenbasierte Flussregel im auf Daten erweiterten DECLARE, welche besagt, dass die Aktivität „Kontoeröffnung“ nicht nach der Aktivität „Beurteilung“ ausgeführt werden darf, wenn das aus „Beurteilung“ ausgehende Datenobjekt „Evaluation“ den Wert „abgelehnt“ aufweist. In obigem Ähnlichkeitsansatz würde diese Regel nicht beachtet werden, da es in gewissen Fällen möglich ist (sofern keine weitere Regel dies einschränkt), „Kontoeröffnung“ nach „Beurteilung“ auszuführen, also $\rightarrow$ (Beurteilung, Kontoeröffnung). Für derartige Regeln scheint es angebracht, direkt auf den jeweiligen Regeln und der zugrunde

Tabelle 4.15: Vollständige Abhängigkeitsmatrix abgleitet aus dem deklarativen Prozessmodell aus Tabelle 4.12.

	A	B	C	D	E	F	G	H
A		$<=$ $\nleftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
B	$=>$ $\leftrightarrow$		$><$ $-$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
C	$\sim$ $\leftrightarrow$	$><$ $-$		$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
D	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$		$^{d}><$ $^{d}-$	$^{d}><$ $^{d}-$	$^{d}><$ $^{d}-$	$><$ $-$
E	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$^{d}><$ $^{d}-$		$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
F	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$^{d}><$ $^{d}-$	$\sim$ $\leftrightarrow$		$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$
G	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$^{d}><$ $^{d}-$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$		$<=$ $\leftrightarrow$
H	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$><$ $-$	$\sim$ $\leftrightarrow$	$\sim$ $\leftrightarrow$	$=>$ $\leftrightarrow$	

liegenden Logik mit Ähnlichkeiten zu arbeiten (siehe Abschnitt 4.5.3.3).

4.6 Einordnung der neu entwickelten Abgleichsmethoden in die identifizierten Anwendungsfelder

Zum Abschluss dieses Kapitels werden die neu entwickelten Abgleichsmethoden in die Anwendungsfeldabbildung 3.7 eingetragen, um mittels der drei Übertragbarkeitsmethoden aus Abschnitt 3.5 die Abdeckung der Anwendungsfelder neu zu beurteilen. Zum Teil sind die Übertragbarkeiten in den jeweiligen Abschnitten bereits angesprochen. Abgleichsmethoden auf Strukturbasis werden, wie in Abschnitt 3.7 geschrieben, nicht weiter betrachtet. Entsprechend fehlt die letzte Zeile in den Abbildungen 4.16 bis 4.19. Das ξ in $VSim^{\xi}$ kann für π, ρ und o stehen. Des Weiteren ist in $VSim^{\xi}$ auch $penVSim^{\xi}$ als Alternativmaß enthalten.

Abbildung 4.16 zeigt die Einordnung der neu entwickelten Abgleichsmethoden in die Anwendungsfelder, für die sie ursprünglich konzipiert sind. Abbildung 4.17 zeigt dann dieselbe Tabelle nach Anwenden der Methodenübertragbarkeit, d. h. nach dem Übertragen der Methoden von imperative auf deklarative Prozessmodelle. Abbildung 4.18 illustriert anschließend die Situation nach Anwenden der Ressourcenübertragbarkeit. Das für die organisatorische Perspektive entwickelte Ähnlichkeitsmaß kann auf die operationale und die datenorientierte Perspektive erweitert werden. Abbildung 4.19 schließlich zeigt die Situation nach der Übertragung aufgrund der Definition der M:N-Abbildung. Alle Methoden sind ebenso für 1:1- bzw. auch für 1:N-Abbildungen anwendbar.

Es stellt sich heraus, dass für alle identifizierten Anwendungsfelder Abgleichsmethoden existieren. Gerade für die Verhaltensperspektive ist eine Auswahl an Methoden verfügbar,

Tabelle 4.16: Abgleichsmatrix für einen Vergleich des imperativen Prozessmodells aus Abbildung 4.12 und des deklarativen Modells aus Tabelle 4.12.

	A	B	C	D	E	F	G	H
A		$=$ $=$	$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $\prec$
B	$\prec$ $\prec$		$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$
C	$\prec$ $\prec$	$=$		$=$ $\prec$	$=$ $\prec$	$=$ $\prec$	$=$ $\prec$	$=$ $\prec$
D	$\prec$ $\prec$	$=$ $\prec$	$=$ $\prec$		$=$ $=$	$=$ $=$	$=$ $=$	$=$ $=$
E	$\prec$ $\prec$	$=$ $\prec$	$=$ $\prec$	$=$ $=$		$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $\prec$
F	$\prec$ $\prec$	$=$ $\prec$	$=$ $\prec$	$=$ $=$	$\prec$ $\prec$		$\prec$ $\prec$	$\prec$ $\prec$
G	$\prec$ $\prec$	$=$ $\prec$	$=$ $\prec$	$=$ $=$	$\prec$ $\prec$	$\prec$ $\prec$		$\prec$ $=$
H	$\prec$ $\prec$	$=$ $\prec$	$=$ $\prec$	$=$ $=$	$\prec$ $\prec$	$\prec$ $\prec$	$\prec$ $=$	

wobei das Maß $VSim^{fdep}$, welches auf Flussabhängigkeiten basiert, keine M:N-Abbildungen unterstützt. Bei einem Vergleich von imperativen und deklarativen Prozessmodellen mit dem Ziel der besseren Verständlichkeit der deklarativen Prozessmodelle (siehe Abschnitt 1.1.3) kann jedoch davon ausgegangen werden, dass dieselben Aktivitäten vorliegen, also dass die Granularität der verglichenen Prozessmodelle dieselbe ist und dann insbesondere auch die Aktivitätsbeschreibungen genau übereinstimmen. Die Ideen für einen Abgleich von deklarativen Prozessmodellen mittels Ausführungspfaden, mittels Verhaltensmustern und auf Logikebene sind in den Tabellen nicht aufgeführt, da sie zum einen ebenfalls nur 1:1-Abbildungen zulassen, zum anderen in ihrer jetzigen Form (noch) zu starke Mängel aufweisen, um sie vernünftig einsetzen zu können. Insbesondere der Abgleich auf der Logik der Regeln bedarf einer eingehenderen Untersuchung, wobei die Anwendbarkeit einer solchen Abgleichsmethode auf deklarative Prozessmodelle, die in der jeweiligen Logik formuliert sind, beschränkt ist. Mehr zu zukünftigen Forschungsfeldern bzw. Erweiterungen findet sich im abschließenden Abschnitt 6.2. Insgesamt ist nun, zusammen mit den Abgleichsmethoden aus den verwandten Arbeiten, von denen einige in Kapitel 2 dargestellt sind, für jeden erwünschten Abgleich mindestens eine Methode verfügbar, wobei in vielen Fällen auch zwischen einer Vielzahl an Methoden gewählt werden kann. Insbesondere kann auch jede Prozessperspektive bei einem Abgleich von imperativen mit deklarativen Prozessmodellen Berücksichtigung finden. Abbildung 4.20 führt noch einmal in Ampeldarstellung auf, inwieweit die Anwendungsfelder mit den in dieser Arbeit entwickelten Methoden nun abgedeckt sind. Zusätzlich zu den Methoden an sich ist auf Basis des vierstufigen Abgleichsansatzes in Abschnitt 4.4 auch ein modulares Vorgehen geschildert, wie ein Abgleich konkret ausgeführt werden kann. Die Auswahl der Ähnlichkeitsmaße hängt von der Modellart, den verwendeten Prozessperspektiven und der Abbildungsart ab.

	imp-imp			dekl-dekl		imp-dekl	
	1:1	1:N	M:N	1:1	M:N	1:1	M:N
funktional			$BSim$				
daten-orientiert							
organi-satorisch			$ASim$				
operational							
verhaltens-orientiert	$VSim^{fdep}$		$VSim^{ord}$ $VSim^{\xi}$				

Abbildung 4.16: Anwendungsfelder und neu entwickelte Methoden.

Die Forschungsfragen, wie sie in Abschnitt 1.4 formuliert sind, können somit als beantwortet angesehen werden. Um die in Abschnitt 1.1 genannten Motive für Ähnlichkeitsabgleiche von Prozessmodellen zu erfüllen, steht für jedes mögliche Anwendungsfeld mindestens ein Ähnlichkeitsmaß zur Verfügung (Abbildungen 4.19 und 4.20). Die Anwendungsfelder decken dabei nicht nur die fünf möglichen Prozessperspektiven ab, sondern berücksichtigen auch unterschiedliche Granulariäten der abzugleichenden Modelle sowie grundlegend die beiden Modellarten, also imperative und deklarative Modelle. Das generelle Abgleichsvorgehen lässt sich in fünf einzeln modifizierbare Module aufteilen, wie in den Abschnitten 3.6 und 4.4.4 erläutert, wobei für das letzte Modul, den Optimierungsschritt des vierstufigen Ansatzes, in Abschnitt 5.2.3 eine Implementierungsmöglichkeit vorgestellt wird.

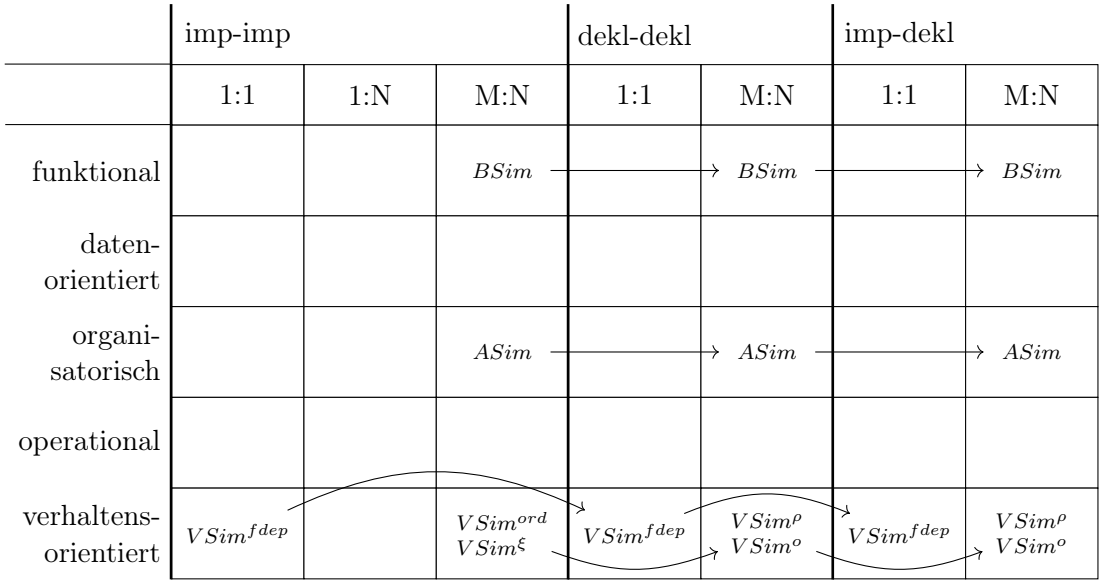


Abbildung 4.17: Anwendungsfelder, neu entwickelte Methoden und Methodenübertragbarkeit.

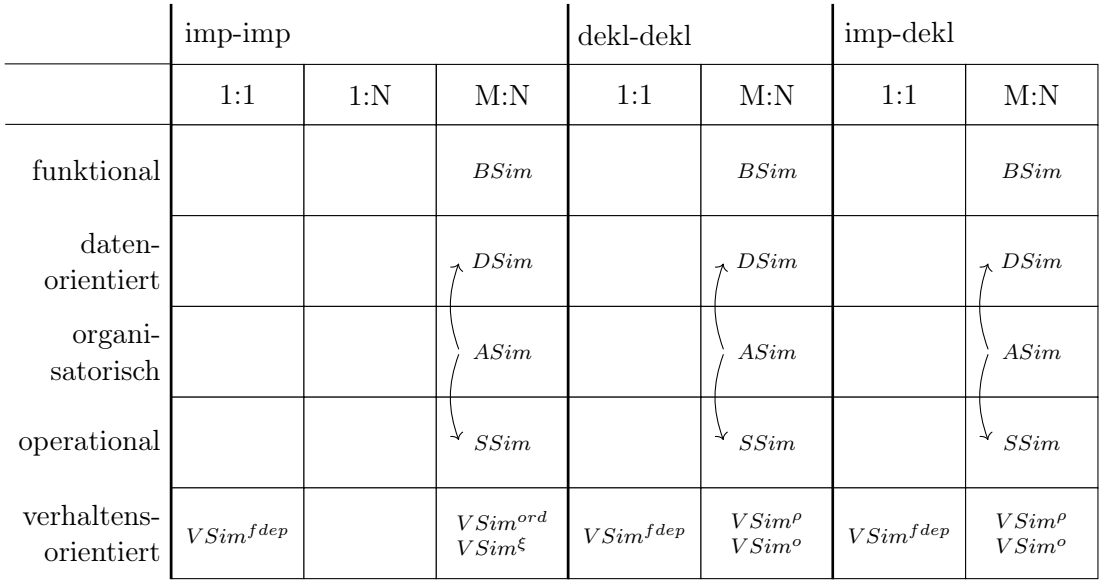


Abbildung 4.18: Anwendungsfelder, neu entwickelte Methoden und Ressourcenübertragbarkeit.

	imp-imp			dekl-dekl		imp-dekl	
	1:1	1:N	M:N	1:1	M:N	1:1	M:N
funktional	$BSim$	$BSim$	$BSim$	$BSim$	$BSim$	$BSim$	$BSim$
daten-orientiert	$DSim$	$DSim$	$DSim$	$DSim$	$DSim$	$DSim$	$DSim$
organi-satorisch	$ASim$	$ASim$	$ASim$	$ASim$	$ASim$	$ASim$	$ASim$
operational	$SSim$	$SSim$	$SSim$	$SSim$	$SSim$	$SSim$	$SSim$
verhaltens-orientiert	$VSim^{fdep}$ $VSim^{ord}$ $VSim^{\xi}$	$VSim^{ord}$ $VSim^{\xi}$	$VSim^{ord}$ $VSim^{\xi}$	$VSim^{fdep}$ $VSim^{\rho}$ $VSim^{o}$	$VSim^{\rho}$ $VSim^{o}$	$VSim^{fdep}$ $VSim^{\rho}$ $VSim^{o}$	$VSim^{\rho}$ $VSim^{o}$

Abbildung 4.19: Anwendungsfelder, neu entwickelte Methoden und Situation nach der Abbildungsübertragbarkeit.

	imp-imp			dekl-dekl		imp-dekl	
	1:1	1:N	M:N	1:1	M:N	1:1	M:N
funktional							
daten-orientiert							
organi-satorisch							
operational							
verhaltens-orientiert							

Abbildung 4.20: Abdeckung der Anwendungsfelder inklusive der neuen Abgleichsmethoden.

Kapitel 5

Implementierung und Evaluation

In diesem Kapitel wird zunächst in Abschnitt 5.1 die zentroidbasierte Verhaltensähnlichkeit basierend auf der Position, Optionalität und Wiederholbarkeit einzelner Aktivitäten separat evaluiert, da dieser Ansatz, das Verhalten als Merkmal der Aktivitäten zu betrachten und nicht über binäre Relationen zu argumentieren, in der Literatur bislang nicht auftaucht. Besonders das Merkmal der Position als Abstraktion der Reihenfolge von Aktivitäten im Prozessmodell soll dabei untersucht werden, um die Güte dieses Maßes besser beurteilen zu können (wie in Abschnitt 4.3.1 diskutiert). Anschließend an diese Einzelevaluation wird in Abschnitt 5.2 eine Proof of Concept-Implementierung vorgestellt, die einige der in Kapitel 4 vorgestellten Abgleichsmethoden umsetzt, unter anderem den zentroidbasierten Verhaltensabgleich und den Vergleich der Ressourcen. Mit Hilfe dieser Implementierung werden die Prozessmodelle, die im Rahmen des Process Model Matching Contests zur Verfügung gestellt werden, untersucht. Der Process Model Matching Contest ist ein Wettbewerb, bei dem verschiedene Abgleichsverfahren gegeneinander antreten um ihre Ergebnisse mit einem verfügbaren Goldstandard zu messen.

5.1 Evaluation der Verhaltensähnlichkeit

Um die Reliabilität des zentroidbasierten Ansatzes, der das Verhalten von Prozessmodellen auf ihre relative Position (Abschnitt 4.3.1), ihre Wiederholbarkeit (Abschnitt 4.3.2) und ihre Optionalität (Abschnitt 4.3.3) hin abstrahiert, zu testen, wird zunächst ein Vergleich dieser Methode mit der der kausalen Fußabdrücke (Abschnitt 2.2.4.4) durchgeführt. Dieser Vergleich findet sich in der Arbeit von Baumann et al. (2015a). Anschließend werden in Abschnitt 5.1.2 einige Abgleichsergebnisse der zentroidbasierten Methode den Einschätzungen von Prozessmodellierungsexperten gegenübergestellt. Die Ergebnisse dieser Befragung sind von Baumann et al. (2016c) veröffentlicht. Diese Einzelevaluation der zentroidbasierten Methode wird durchgeführt, da die Kombination der drei genannten Merkmale zur Beschreibung des Verhaltens völlig neu ist. Die positiven Ergebnisse dieser Einzelevaluation bestätigen dann den Einsatz des zentroidbasierten Ansatzes in der Proof of Concept-Implementierung (Abschnitt 5.2).

5.1.1 Validierung des zentroidbasierten Ansatzes

Um die Ergebnisse des zentroidbasierten Ansatzes einordnen zu können, wird ein Vergleich mit der Methode der kausalen Fußabdrücke (siehe Abschnitt 2.2.4.4) und der vereinfachten Variante davon, der Methode der kleinsten kausalen Fußabdrücke, die von Becker und Laue

(2012) vorgestellt wird, durchgeführt. Hierzu werden die drei Prozessmodelle aus den Abbildungen 5.1, 5.2 und 5.3 miteinander abgeglichen.

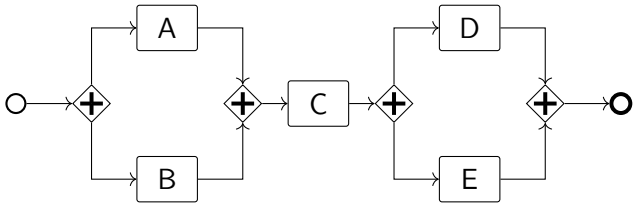


Abbildung 5.1: Beispielmodell G_1 zur Validierung des zentroidbasierten Ansatzes.

Abbildung 5.2: Beispielmodell G_2 zur Validierung des zentroidbasierten Ansatzes.

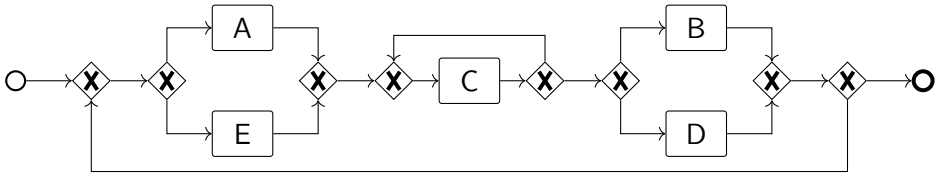


Abbildung 5.3: Beispielmodell G_3 zur Validierung des zentroidbasierten Ansatzes.

Die Prozessmodelle G_1 und G_2 beschreiben den gleichen Prozess, die Modelle wurden lediglich von verschiedenen Personen erstellt, die die zu erledigenden Aufgaben A bis E unterschiedlich auf die Aktivitäten aufgeteilt haben. Modell G_3 beschreibt einen anderen Prozess, allerdings mit ähnlichen Beschreibungen der Aktivitäten. Die Aufgabenbeschreibungen sind für den Abgleich durch Buchstaben ersetzt.

Für die Validierung wird jeweils eine Abbildung zwischen den Beispielmodellen G_1 und G_2 bzw. zwischen G_1 und G_3 fest vorgegeben. Die bijektive M:N-Abbildung zwischen G_1 und G_2 sei $M^b_{1,2} = \{(\{A, B\}, \{AB\}), (\{C\}, \{C\}), (\{D, E\}, \{DE\})\}$, d. h., die Abbildung bildet die Aktivitäten mit gleicher (konkatenierter) Aufgabenbeschreibung aufeinander ab. Für die (kleinsten) kausalen Fußabdrücke kann jedoch nur eine partiell injektive 1:1-Abbildung zwischen den Modellen gebildet werden, diese ist in diesem Fall $M^{pi}_{1,2} = \{(A, AB), (C, C), (D, DE)\}$, d. h., auch hier werden möglichst Aktivitäten mit gleicher Beschreibung aufeinander abgebildet, wobei wegen der Einschränkung auf 1:1-Abbildungen die Aktivitäten B und E aus G_1 gelöscht werden. Statt B hätte auch A oder statt E auch D gelöscht werden können. Bezogen allein auf die Levenshtein-Ähnlichkeit (siehe Abschnitt 2.2.2.1) liefern $M^b_{1,2}$ und $M^{pi}_{1,2}$ den jeweils größten Ähnlichkeitswert bzw. einen größten Wert. Um das Löschen der Aktivitäten bzw. die nur teilweise Übereinstimmung der Aktivitäten bei der 1:1-Abbildung zu berücksichtigen, wird die von Dijkman et al. (2011) vorgeschlagene Methode, die die Ähnlichkeit der Aufgabenbeschreibungen bei der Berechnung der Fußabdruck-Ähnlichkeit einbezieht, verwendet. Hierbei ist die Ähnlichkeit der Zeichenketten „A“ und „AB“ bzw. „D“ und „DE“ jeweils 0,5.

Die Abbildung zwischen G_1 und G_3, die bezogen auf die Aktivitätenbeschreibungen den höchsten Ähnlichkeitswert liefert, ist die Abbildung, die jeder Aktivität die mit der gleichen Beschreibung im anderen Modell zuordnet. Die bijektive M:N-Abbildung ist in diesem Fall

$M^b_{1,3} = \{(\{A\},\{A\}),(\{B\},\{B\}),(\{C\},\{C\}),(\{D\},\{D\}),(\{E\},\{E\})\}$ und die partiell injektive 1:1-Abbildung $M^{pi}_{1,3} = \{(A,A),(B,B),(C,C),(D,D),(E,E)\}$.

Für den Abgleich von G_1 und G_2 mit dem zentroidbasierten Ansatz inklusive Straftermen ergibt sich ein Ähnlichkeitswert von

$$\begin{aligned}
&GSim_{M^b_{1,2}}(G_1, G_2) \\
&= \frac{1}{3}\left(penVSim^{\pi}(G_1, G_2) + penVSim^{\rho}(G_1, G_2) + penVSim^{o}(G_1, G_2)\right) \\
&= 1,000.
\end{aligned}$$

Der Wert, der sich für den Abgleich der kausalen Fußabdrücke ergibt, ist

$$fsim_{cfp}(G_1, G_2) = 0,799.$$

Für den Abgleich von G_1 und G_3 mit dem zentroidbasierten Ansatz inklusive Straftermen ergibt sich ein Ähnlichkeitswert von

$$\begin{aligned}
&GSim_{M^b_{1,3}}(G_1, G_3) \\
&= \frac{1}{3}\left(penVSim^{\pi}(G_1, G_3) + penVSim^{\rho}(G_1, G_3) + penVSim^{o}(G_1, G_3)\right) \\
&= 0,333.
\end{aligned}$$

Der Wert, der sich für den Abgleich der kausalen Fußabdrücke ergibt, ist

$$fsim_{cfp}(G_1, G_3) = 0,640.$$

Die Werte für die Methode der kleinsten kausalen Fußabdrücke liegen jeweils dazwischen, wie in Tabelle 5.1 dargestellt. Zusätzlich sind in der Tabelle noch die jeweiligen Anzahlen der berechneten Zwischenwerte (keine elementaren arithmetischen Operationen) bei der Kalkulation des Ähnlichkeitswerts angegeben. Diese sind mit deutlichem Abstand für das normale Verfahren der kausalen Fußabdrücke am höchsten und werden mit der Methode der kleinsten kausalen Fußabdrücke, wie von Becker und Laue (2012) behauptet, deutlich verringert. Steigt die Anzahl der berechneten Zwischenwerte für die kausalen Fußabdrücke mit der Anzahl an Aktivitäten exponentiell an, so ist der Anstieg bei den kleinsten kausalen Fußabdrücken nur noch quadratisch. Der zentroidbasierte Ansatz induziert einen linearen Anstieg der berechneten Zwischenwerte mit der Anzahl an Aktivitäten. Voraussetzung dafür ist allerdings, dass die relative Position, die Optionalität und die Wiederholbarkeit bekannt sind. Vor allem die automatisierte Bestimmung der Optionalität ist, wie bei der Implementierung in Abschnitt 5.2.1.2 ersichtlich, nicht trivial.

Alle drei Verfahren schätzen die Ähnlichkeit von G_1 zu G_2 größer ein als die Ähnlichkeit zwischen G_1 und G_3. Der zentroidbasierte Ansatz spricht dem Paar (G_1, G_2) sogar eine Gleichheit des Verhaltens zu. Die parallelen Verzweigungen werden hier vollständig ignoriert. Bezogen auf das Ergebnis nach der Ausführung, nämlich dass alle fünf einzelnen Aktivitäten bzw. teilweise aggregierten Aktivitäten am Ende genau einmal ausgeführt sind, ist dies auch als nicht problematisch einzuschätzen. Es wird lediglich die Unabhängigkeit der Aktivitäten A und B bzw. D und E und die Beliebigkeit der Reihenfolge ihrer Ausführung ignoriert. Durch die Zusammenlegung der Beschreibungen in den Aktivitäten AB und DE in G_2 scheint eine Reihenfolge implizit gegeben zu sein, wobei der Grund für die Konkatenation in dieser Reihenfolge aus dem Modell nicht ersichtlich ist.

Tabelle 5.1: Ähnlichkeitswerte für die Methode der kausalen Fußabdrücke (CF), der kleinsten kausalen Fußabdrücke (sCF) und des zentroidbasierten Ansatzes (CB); zusätzlich ist jeweils noch die Anzahl der berechneten Zwischenwerte (#IM) angegeben.

$Sim.\ (\#IM)$	CF	sCF	CB
$Sim(G_1, G_2)$	0.799 (294)	0.885 (90)	1.000 (30)
$Sim(G_1, G_3)$	0.640 (414)	0.632 (108)	0.333 (30)

Bezogen auf den Abgleich von G_1 und G_3 weist die Methode der (kleinsten) kausalen Fußabdrücke den Modellen unter der gegebenen Abbildung einen Ähnlichkeitswert von beinahe $^2/_3$ zu, und das, obwohl die Ausführungsmöglichkeiten sich sehr voneinander unterscheiden. Im Modell G_3 sind alle Aktivitäten wiederholbar und außer C muss keine der Aktivitäten für sich betrachtet überhaupt ausgeführt werden. Der zentroidbasierte Ansatz vergibt hier einen Ähnlichkeitswert von $^1/_3$, der deutlich unter dem oft verwendeten Schwellenwert von 0,5 liegt (siehe Gleichung (2.3)). Die Modelle G_1 und G_3 würden vom zentroidbasierten Ansatz also nicht als ähnliche Kandidaten erkannt werden, während die Methode der (kleinsten) kausalen Fußabdrücke nicht nur G_1 und G_2 sondern auch diese beiden Modelle als ähnlich einstuft. Es lässt sich also feststellen, dass die Grade der Ähnlichkeit (deren relative Werte) vom zentroidbasierten Ansatz und von der Methode der kausalen Fußabdrücke gleich eingeschätzt werden. Jedoch scheinen, aus subjektiver Sicht, die absoluten Ähnlichkeitswerte, die der zentroidbasierte Ansatz ausgibt, plausibler im Vergleich zu denen der kausalen Fußabdrücke. Außerdem ist der Rechenaufwand des zentroidbasierten Ansatzes deutlich geringer als der der anderen beiden Methoden mit kausalen Fußabdrücken, weshalb zumindest nach diesen ersten Ergebnissen der zentroidbasierte Ansatz dem der kausalen Fußabdrücke (in beiden Formen) vorzuziehen ist.

Da der Vergleich des zentroidbasierten Ansatzes mit der Methode der kausalen Fußabdrücke unterschiedliche Rahmenbedingungen voraussetzt, also dass die kausalen Fußabdrücke nur 1:1-Abbildungen zulassen, ist der Vergleich der beiden bzw. der drei Methoden nur ein erster Anhaltspunkt für das Abschneiden des zentroidbasierten Ansatzes. Es existieren jedoch keine vergleichbaren Messmethoden der Verhaltensähnlichkeit für M:N-Abbildungen. Deswegen wird im folgenden Abschnitt ein Vergleich mit einer von Experten eingeschätzten Ähnlichkeit unter M:N-Abbildungen durchgeführt. Diese Evaluation findet sich in der Arbeit von Baumann et al. (2016c).

5.1.2 Vergleich mit Experteneinschätzung

Für das Abschneiden des zentroidbasierten Ansatzes im Vergleich zur Meinung von Prozessmodellierungsexperten werden, wie von Baumann et al. (2016c) erläutert, zu fünf Prozessmodellen jeweils drei Varianten gezeigt, wobei unter einer gegebenen M:N-Abbildung anzugeben ist, welche Variante bezogen auf das Verhalten dem Referenzmodell am ähnlichsten und welche am unähnlichsten ist. Das erste Referenzmodell mit seinen drei Varianten ist in Abbildung 5.4 gezeigt, das letzte in Abbildung 5.5. Die übrigen drei Modellbeispiele finden sich in Anhang A.4. Zu einer Menge zusammengefasste Aktivitäten sind jeweils gleich gemustert. Die Aktivitäten der Bildmenge weisen das gleiche Muster auf. Ein zusätzlicher Vergleich mit Werten, die andere Ähnlichkeitsmaße liefern, findet hier im Unterschied zu Abschnitt 5.1.1

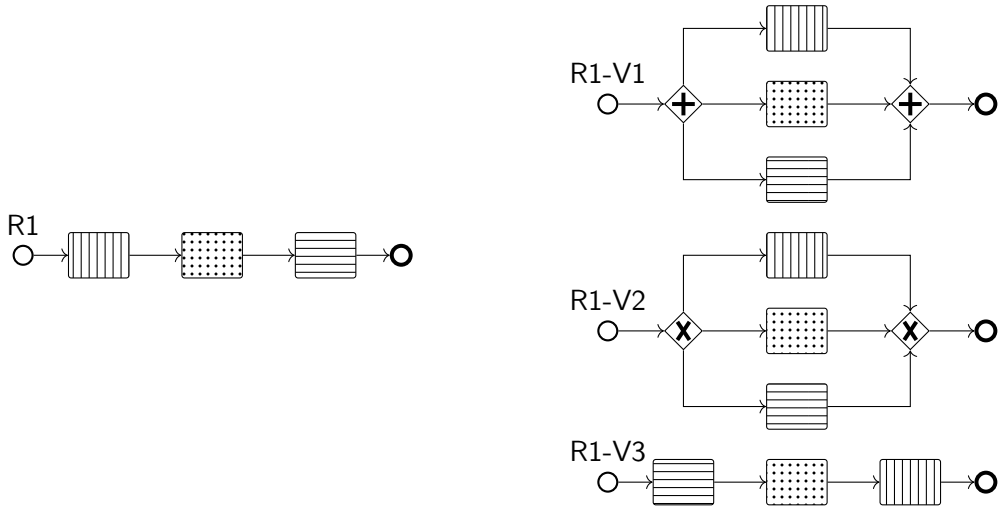


Abbildung 5.4: Referenzmodell 1 (links) und drei Varianten (von oben nach unten Variante 1 bis 3).

nicht statt, da explizit auch M:N-Abbildungen in den Referenzmodellen vorliegen und für diese in der Literatur bislang keine Ähnlichkeitsmaße existieren. Aus diesem Grund wird der Vergleich mit einer Experteneinschätzung durchgeführt.

Tabelle 5.2 fasst die Ergebnisse der Umfrage zusammen und gibt die Ähnlichkeitswerte an, die der zentroidbasierte Ansatz liefert. Im Vergleich zu den Ergebnissen, die von Baumann et al. (2016c) präsentiert werden, unterscheiden sich die Zahlen etwas, da seit der Veröffentlichung der entsprechenden Arbeit weitere Teilnahmen an der Umfrage erfolgt sind. Bis auf Referenzmodell 5 ist es immer der Fall, dass dasjenige Modell, das von den Experten als das ähnlichste zum Referenzmodell eingeschätzt wird, auch bei der Berechnung der Ähnlichkeitswerte den höchsten Wert erzielt. Somit spiegelt die automatisierte Ähnlichkeitsberechnung bis auf Referenzmodell 5 die Meinung der Experten wider und kann trotz des einen Ausreißers, der nachfolgend noch diskutiert wird, als gutes und vor allem auch einfach zu berechnendes Maß für die Verhaltensähnlichkeit von Prozessmodellen angesehen werden.

Die Diskrepanz in Referenzmodell 5 lässt sich dadurch erklären, dass die Länge von R5 und R5-V1 und damit auch die relativen Positionen deutlich ähnlicher sind als die von R5 und R5-V3. Der zentroidbasierte Ansatz unterschiedet keine verschiedenartigen Schleifen, d. h., ob die Aktivitäten einzeln wiederholbar sind (wie in R5 und in R5-V3) oder nur zusammen wiederholbar sind (wie in R5-V1), wirkt sich nicht auf den Ähnlichkeitswert aus. Diese Tatsache dürfte aber der Grund sein, warum die Modellvariante R5-V3 von den Experten als ähnlicher zum Referenzmodell eingeordnet wird. Insgesamt sind die Ähnlichkeitswerte von R5-V1 und R5-V3, die der zentroidbasierte Ansatz ausgibt, ziemlich hoch und unterschieden sich nicht stark voneinander. Das Modell R5 und seine Varianten wurden genau aus dem Grund, um den Effekt der einzelnen und gemeinsamen Wiederholbarkeiten zu testen und die Grenzen des Verhaltensähnlichkeitsansatzes aufzuzeigen, konstruiert. Die unähnlichsten Modelle wurden vom zentroidbasierten Ansatz und von den Experten, wieder mit einer Ausnahme im Fall von Modell 5, übereinstimmend bewertet. Im Fall 5 ist von den Experten mehrheitlich kein unähnlichstes Modell genannt, vielmehr sind jeweils knapp 40% der Stimmen zwischen den Varianten R5-V1 und R5-V2 zu gleichen Teilen aufgeteilt. Da diese beiden Modelle dieselbe, anders als im Referenzmodell modellierte Schleife aufweisen, nämlich eine Schleife, die im-

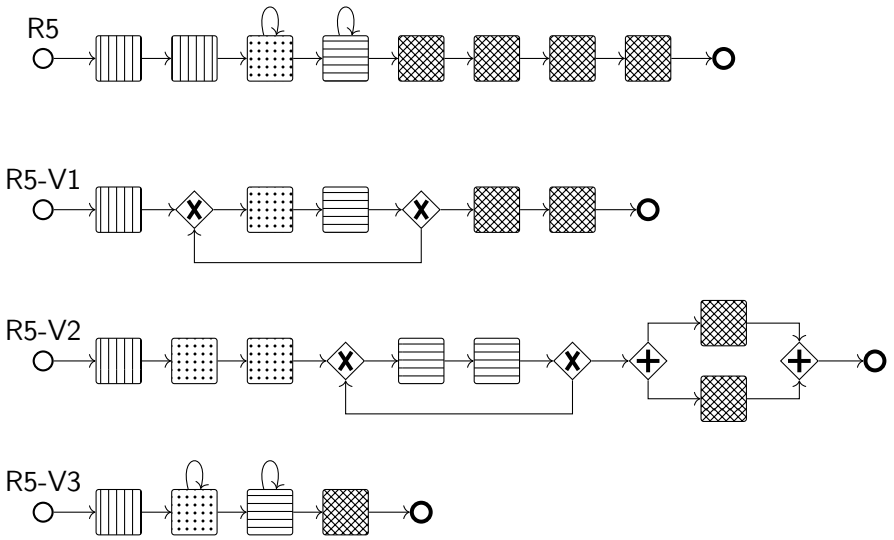


Abbildung 5.5: Referenzmodell 5 (oben) und drei Varianten darunter (von oben nach unten Variante 1 bis 3).

mer nur eine gemeinsame Wiederholung der beteiligten Aktivitäten erlaubt, scheint dies der Ausschlag für die zugewiesene Unähnlichkeit zu sein. Es ist also zu überlegen, ob zusätzlich zu den Wiederholbarkeits- und Optionalitätsangaben für die einzelnen Aktivitäten auch eine gemeinsame Angabe für eine Menge von Aktivitäten, möglicherweise über einen Block-Ansatz wie von Bae et al. (2006a), angebracht ist.

Tabelle 5.2: Prozentuale Werte des unbestraften, zentroidbasierten Ansatzes (CB; Ähnlichkeitswert jeweils aus [0,100]) und Umfragedaten (pro Variante in Summe 100).

		1-1	1-2	1-3	2-1	2-2	2-3	3-1	3-2	3-3	4-1	4-2	4-3	5-1	5-2	5-3
	CB	94	61	89	83	100	75	85	89	48	65	91	97	99	93	98
Umfrage	sim	61,5	7,7	30,8	0	100	0	7,7	92,3	0	30,8	0	69,2	23,1	0	76,9
	dissim	0	84,6	15,4	7,7	0	92,3	15,4	7,7	76,9	69,2	30,8	0	38,5	38,5	23,1

5.2 Proof of Concept-Implementierung

Die prototypische Implementierung des Abgleichsansatzes, der die unten angegebenen Einzelähnlichkeiten in der Zielfunktion des vierstufigen Ansatzes verwendet, basiert auf Python[1] und einem ganzzahligen linearen Programm geschrieben in der Modellierungssprache ZIMPL[2]. Das ganzzahlige lineare Programm modelliert das Optimierungsproblem des Findens einer besten Abbildung[3] und wird mit der SCIP Optimization Suite[4] gelöst. Als Eingabe

[1] Es wird hierfür die Version 2.7 verwendet, verfügbar auf `https://www.python.org/downloads/` [letzter Zugriff: 14.06.2017]

[2] `http://zimpl.zib.de/` [letzter Zugriff: 16.06.2017]

[3] Da es grundsätzlich möglich ist, dass mehr als eine Optimallösung existiert, wird hier von *einer* besten Abbildung und nicht von *der* besten Abbildung gesprochen.

[4] `http://scip.zib.de/` [letzter Zugriff: 16.06.2017]

sind BPMN-Modelle zulässig. Grob gliedert sich die Implementierung in drei Teile. Der erste Teil verarbeitet die eingehenden BPMN-Modelle vor und weist den einzelnen Aktivitäten die für den Ähnlichkeitsabgleich relevanten Eigenschaften und Merkmale zu. Der zweite Teil berechnet die Ähnlichkeiten zwischen den Mengen an Aktivitäten, wobei hierfür zunächst die Potenzmenge der Aktivitäten gebildet wird. Der dritte Teil nutzt die ganzzahlige, lineare Optimierung, um eine beste Abbildung, d. h. eine mit dem größten Ähnlichkeitswert, zu finden. Die ersten beiden Teile, die auf Python basieren, entstanden in Zusammenarbeit mit Marcel Bankau und Michael Baumann, der dritte Teil, das ZIMPL-Programm, in Zusammenarbeit mit Susanne Hoffmeister und Jörg Rambau. Am konzeptuellen Hintergrund haben die Mitarbeiter des Lehrstuhls AI4, Lars Ackermann und Stefan Schönig, sowie der Lehrstuhlinhaber Stefan Jablonski mitgewirkt. Als Ähnlichkeitswerte werden die Beschreibungsähnlichkeit $BSim$ (Abschnitt 4.1), die Agentenähnlichkeit $ASim$ (Abschnitt 4.2.1.2), die Dokumentenähnlichkeit $DSim$ (Abschnitt 4.2.2), die Positionsähnlichkeit $VSim^{\pi}$ (Abschnitt 4.3.1), die Wiederholbarkeitsähnlichkeit $VSim^{\rho}$ (Abschnitt 4.3.2) und die Optionalitätsähnlicheit $VSim^{o}$ (Abschnitt 4.3.3) bzw. deren bestrafte Versionen $penVSim^{\pi}$, $penVSim^{\rho}$ und $penVsim^{o}$ (Abschnitt 4.3.4) sowie die Knotenähnlichkeit $NSim$ und die Kantenähnlichkeit $ESim$ (Abschnitt 4.4.3) verwendet. An einigen wenigen Stellen müssen die Ausgaben des Python-Programms zur Eingabe in das ZIMPL-Programm manuell leicht angepasst werden, da sich in ZIMPL die modifizierten Eingaben schneller einlesen lassen. Für eine zukünftige Weiterarbeit mit dem Programm bzw. bei einer Integration der verschiedenen Teile ist diese Modifikation im Python-Code noch einzuarbeiten.

5.2.1 Vorverarbeitung der Prozessmodelle

Im Vorverarbeitungsschritt werden die zu vergleichenden Prozessmodelle eingelesen, für Aktivitäten und Events die zugehörigen Ressourcenmengen (Agenten und Datenobjekte) sowie die Merkmale Position, Wiederholbarkeit und Optionalität für den verhaltensbasierten Abgleich aus Abschnitt 4.3 bestimmt und anschließend mehrere CSV-Dokumente erzeugt, die für den zweiten und dritten Schritt benötigt werden.

5.2.1.1 Eingehende BPMN-Modelle

Die Implementierung ist für BPMN-Modelle ausgelegt, die mittels des Camunda Modelers[5] erstellt wurden. Auch wenn BPMN zwar eine vorgegebene, standardisierte Syntax hat, so werden in verschiedenen Modellierumgebungen doch teilweise andere XML-Tags benutzt, sodass das korrekte Parsen der BPMN-Modelle dann nicht mehr möglich ist.

Wegen der Abbildbarkeit der BPMN-Modelle auf Prozessgraphen gemäß Definition 3.1, werden vom Parser nicht alle Modellelemente, die möglich wären, unterstützt. Aktivitäten sind in ihrer einfachsten Form ohne Markierungen oder Typen zu verwenden. Insbesondere sind so auch keine Subprozesse möglich. Verzweigungen können parallel (AND-Gateway), exklusiv (XOR-Gateway) oder inklusiv (OR-Gateway) sein, wobei das OR-Gateway bei der Bestimmung der verhaltensrelevanten Informationen wie ein XOR-Gateway behandelt wird, da gegenseitige Ausschließlichkeit von Aktivitäten bei den betrachteten Verhaltensmerkmalen (Position, Optionalität, Wiederholbarkeit) nicht berücksichtigt wird. Events werden grundsätzlich wie Aktivitäten behandelt, wobei entweder ihre Anmerkungen oder, wenn keine Anmerkungen vorhanden sind, ihr jeweiliger Typ als Beschreibung verwendet wird. Sequenzflüsse und auch Nachrichtenflüsse sind zulässig. Abbildung 5.6 zeigt ein BPMN-Modell, das im

[5] Zu finden auf `https://camunda.org/bpmn/tool`, letzter Zugriff: 14.06.2017

Weiteren als Beispiel für die verschiedenen Dokumente, die während des Programmdurchlaufs erzeugt werden, dient.

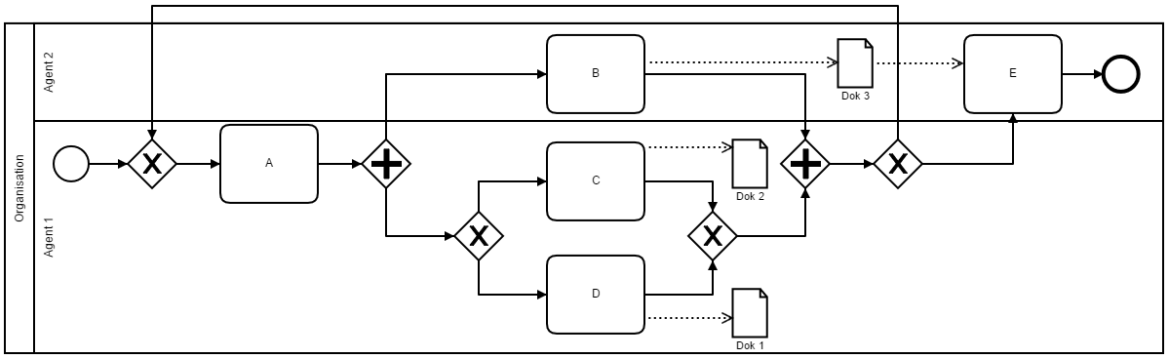


Abbildung 5.6: BPMN Beispielmodell zur Illustration der Implementierung

5.2.1.2 Einlesen der Modelle und Festlegen der Knotenmerkmale

Beim Einlesen der Modelle werden zunächst alle Knoten des Prozessmodells, das sind das Start- und Endereignis, die Aktivitäten, die Zwischenereignisse und die Gateways, erkannt. Für die erkannten Aktivitäten und Ereignisse wird eine Merkmalsliste initialisiert, die zu jedem dieser Knoten folgende Informationen bereitstellt:

- ID: Vom Camunda-Modeler vergebene ID, die direkt aus der XML-Datei des Prozessmodells gelesen wird.
- Typ: Der Typ des Knotens (Start-/Endereignis, Zwischenereignis, Aktivität).
- Beschreibung: Bei Aktivitäten ist das die Aufgabenbeschreibung, bei Ereignissen die Beschreibung oder, falls keine angegeben ist, der Typ des Ereignisses (z. B. *StartEvent*).
- Vorgängerknoten: Eine Liste aller direkten Vorgängerknoten (Aktivitäten und Ereignisse).
- Nachfolgerknoten: Eine Liste aller direkten Nachfolgerknoten (Aktivitäten und Ereignisse).
- Eingehende Datenobjekte: Eine Liste aller eingehenden Datenobjekte (Artifacts).
- Ausgehende Datenobjekte: Eine Liste aller ausgehenden Datenobjekte (Artifacts).
- Agenten: Zugeordneter Agentenname der Lane (falls vorhanden) oder des Pools.
- Services: (bleibt leer)
- Optionalität: Boolscher Wert, der mit Hilfe des Sequenzflusses und der Gateways bestimmt wird.
- Position: Relative Position (zwischen 0 und 1), die mit Hilfe des Sequenzflusses und der Start- und Endereignisse bestimmt wird.
- Wiederholbarkeit: Boolscher Wert, der mit Hilfe des Sequenzflusses bestimmt wird.

Gateways werden ebenfalls erkannt, allerdings werden sie, da sie für die Abbildung nicht benötigt werden, nicht in die Merkmalsliste aufgenommen. Ereignisse werden wie Aktivitäten behandelt, also auch in der Abbildung berücksichtigt, wobei für das Start- und Endereignis die Abbildungsvorschrift festgelegt ist, weswegen diese beiden Paare auch nicht explizit in der Abbildung auftauchen. Nachrichtenflüsse werden wie Sequenzflüsse behandelt. Insbesondere sind Vorgänger- und Nachfolgerknoten auch über Nachrichtenflüsse zu bestimmen.

Bestimmen der relativen Position Die relative Position eines Knotens wird dadurch bestimmt, dass die Länge des kürzesten Pfads vom Startereignis zum jeweiligen Knoten geteilt wird durch ebendiese Länge plus der Länge des kürzesten Pfades vom Knoten zum (nächstgelegenen) Endereignis. Gateways werden hierbei nicht mitgezählt, d. h., es wird eine Abstraktion dieser Knoten wie in Abgleichsmethode 1 aus Abschnitt 2.2.3.4 durchgeführt. Da Schleifen in den Modellen erlaubt sind, ist eine Abfrage, ob ein Pfad bei der Traversierung des Graphen schon einmal besucht wurde, notwendig, um Endlossuchen zu vermeiden. Zur Bestimmung des kürzesten Pfades wird nach dem Algorithmus von Dijkstra (1959) vorgegangen, wobei alle Kanten gleich gewichtet werden.

Bestimmen der Wiederholbarkeit Zur Bestimmung der Wiederholbarkeit eines Knotens wird eine Tiefensuche vom jeweiligen Knoten aus gestartet. Befindet sich der Knoten selbst, außer direkt bei Start, auf einem Pfad dieser Suche, so ist er wiederholbar. Sobald ein Knoten als wiederholbar erkannt wird, wird die Suche abgebrochen.

Bestimmen der Optionalität Die Bestimmung optionaler Knoten wird dadurch erschwert, dass für die Prozessmodelle keine Blockstruktur vorausgesetzt wird, dass also für die in dieser Arbeit zugelassenen Modelle nicht für jedes Split-Gateway genau ein korrespondierendes Join-Gateway existiert und umgekehrt, wobei es genau so viele Split- wie Join-Gateways gibt. Ein Knoten ist optional, wenn es einen Ausführungspfad vom Start- zum Endknoten gibt, der den Knoten nicht mit einschließt. Andersherum ist ein Knoten verpflichtend (nicht optional), wenn jeder Ausführungspfad vom Start- zum Endknoten diesen Knoten einschließt. Die Optionalitätsbestimmung wird über eine sukkzessive Markierung der Knoten vorgenommen, wobei schrittweise wie folgt vorgegangen wird:

1. Beginnend am Startknoten wird eine Startmarkierung (tag=1) an jedem darauffolgenden Knoten angebracht, bis ein (X)OR-Split erreicht wird.

2. Alle ausgehenden Pfade des (X)OR-Splits erhalten jeweils eine neue Markierung, die die alte Markierung erweitert, wobei jeder ausgehende Pfad eine unterschiedliche Erweiterung erhält (tag=1.1 für den ersten ausgehenden Pfad, tag=1.2 für den zweiten ausgehenden Pfad usw.). Die Reihenfolge ist hierbei unerheblich.

3. Trifft eine Markierung, ausgelöst durch Schleifen im Modell, auf bereits markierte XOR-Splits, muss dort (und nur dort) eine Fallunterscheidung durchgeführt werden:

   - Die aktuelle Markierung ist in den ausgehenden Pfaden des Splits genau so schon einmal enthalten (z. B. aktueller tag=1.2, vorhandener tag=1.2) bzw. ein Teilstring der aktuellen Markierung, von Beginn ab getestet, ist in den ausgehenden Pfaden des Splits enthalten (z. B. aktueller tag=1.2, vorhandener tag=1): Erzeuge keine Erweiterung der aktuellen Markierung sondern schiebe sie unverändert in alle ausgehenden Pfade weiter (wo sie noch nicht auftaucht).

- Ist die vorhergehende Bedingung nicht erfüllt, dann erweitere die aktuelle Markierung (wie in 2.) und schiebe sie in alle ausgehenden Pfade weiter.

4. Fahre so lange fort, bis jede Markierung bzw. jeder Pfad bei einem Endereignis angekommen ist. Alle Knoten sind nun mit mindestens einer Markierung versehen.

5. Für jeden Knoten wird, beginnend mit den längsten Markierungen, getestet, ob alle parallelen Markierungen eines (X)OR-Splits vorhanden sind. Falls ja, werden diese Markierungen alle entfernt und durch die (gemeinsame) Vorgängermarkierung, die um eins kürzer ist, ersetzt (z. B. tag=1.1 und tag=1.2 werden, falls der zugehörige (X)OR-Split genau zwei ausgehende Pfade hat, zu tag=1 verschmolzen). Dies wird für jeden Knoten so lange durchgeführt, bis keine Ersetzungen mehr vorgenommen werden können.

6. Jeder Knoten, der unter anderem die Startmarkierung (tag=1) hat, ist nicht optional; alle anderen Knoten, d. h. die, die nicht die Startmarkierung haben, sind optional. Für diese gibt es einen Weg vom Startknoten zum Endknoten, ohne über den betrachteten Knoten (ohne Startmarkierung) zu gehen.

Für die Bestimmung der Optionalität werden Informationen der Gateways, insbesondere der XOR- und OR-Gateways, benötigt. Das heißt, dass die Gateways, die für die Abbildung keine Rolle spielen, erst nach diesem Schritt aus den Berechnungen entfernt werden. In Anhang A.2 ist ein beispielhafter Durchlauf des Algorithmus zur Optionalität von Knoten zu finden.

Der Algorithmus zum Verteilen der Markierungen terminiert, da die Anzahl der (X)OR-Splits im Modell endlich ist. Da Markierungen nur erweitert werden, wenn ein Pfad auf einen (X)OR-Split trifft, den er zuvor noch nicht besucht hat (wenn die Markierung oder ein Teil davon noch nicht vorhanden ist), kann eine Markierung maximal nur so oft erweitert werden, d. h., eine Markierung kann maximal so lang werden, wie es (X)OR-Splits gibt. Ein verpflichtender (nicht optionaler) Knoten liegt auf jedem Ausführungspfad vom Start- zum Endknoten. Das heißt, zu jedem anderen Knoten auf einem Ausführungspfad kann festgelegt werden, ob dieser vor der ersten Ausführung des betrachteten Knotens oder nach dieser liegt. Das bedeutet aber, dass jeder (X)OR-Split, der vor dem betrachteten Knoten liegt, auch einen (X)OR-Join vor diesem Knoten hat, bei dem die Markierungen des (X)OR-Splits aufeinander treffen. Wäre dies nicht so, gäbe es Pfade, auf denen der betrachtete Knoten nicht liegt, was jedoch nicht möglich ist, da der Knoten verpflichtend ist. Andersherum muss jeder Knoten, der alle entscheidenden Markierungen hat, nämlich solche, die sich zur Startmarkierung verschmelzen, zwingend ausgeführt werden, da Optionalität nur mittels (X)OR-Splits entstehen kann und alle bis zu einem bestimmten Zeitpunkt aus (X)OR-Splits ausgehenden Pfade bei einem Knoten mit der Startmarkierung zuvor wieder zusammengelaufen sind, sonst wären nicht alle entscheidenden Markierungen an diesem Knoten vorhanden.

5.2.1.3 Erstellen der Merkmalsliste und der Knoten- und Kantenliste

Nach der Bestimmung bzw. Berechnung der festgelegten Knotenmerkmale wird eine Merkmalsliste, eine CSV-Datei, erzeugt. Abbildung 5.7 zeigt einen Ausschnitt aus der Merkmalsliste des Prozessmodells aus Abbildung 5.6. In der ersten Zeile dieser Liste muss ein Abgleich der Agenten, die aus den Prozessmodellen extrahiert wurden, manuell vorgenommen werden. Gleichen Agenten bzw. Rollen wird hierbei dieselbe (beliebige) Nummer zugewiesen.

Zusätzlich zur Merkmalsliste werden eine CSV-Datei mit den IDs aller Knoten und eine CSV-Datei mit den IDs aller Kanten ausgegeben. Diese beiden Dateien werden im dritten Schritt des Algorithmus, also im Schritt zur Bestimmung der besten Abbildung, benötigt, nämlich dann, wenn der Anteil der abgebildeten Knoten und abgebildeten Kanten unter einer bestimmten Abbildung berechnet wird.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Organisation		END									
2	id	type	desc	in nodes	out nodes	in data	out data	agent	services	opt	pos	rep
3	Task_0x3fuvs	task	D	Task_0iph5v2	Task_1jchn9e Task_0iph5v2		DataObjectReference_0lw4n65	Organisation		1	0.5	1
4	Task_0iph5v2	task	A	StartEvent_1 Task_1cerqes Task_0x3fuvs Task_1rk2fxq	Task_1cerqes Task_1rk2fxq Task_0x3fuvs			Organisation		0	0.25	1
5	Task_1rk2fxq	task	C	Task_0iph5v2	Task_1jchn9e Task_0iph5v2		DataObjectReference_0ikoser	Organisation		1	0.5	1
6	Task_1jchn9e	task	E	Task_1cerqes Task_0x3fuvs Task_1rk2fxq	EndEvent_1g3bgnj	DataObjectReference_1i1tfx9		Organisation		0	0.75	0
7	Task_1cerqes	task	B	Task_0iph5v2	Task_1jchn9e Task_0iph5v2		DataObjectReference_1i1tfx9	Organisation		0	0.5	1

Abbildung 5.7: Merkmalsliste als CSV-Datei des Beispielsmodells.

5.2.2 Aufstellen der Ähnlichkeitsmatrix

Im zweiten Schritt der Implementierung wird eine Ähnlichkeitsmatrix aufgestellt, in der für zwei Knotenmengen eine Reihe bestimmter Ähnlichkeiten gelistet ist. Es werden hierfür die Merkmalslisten der zwei zu vergleichenden Prozessmodelle benötigt.

5.2.2.1 Aufstellen der Potenzmengen

Für jedes Modell wird die Potenzmenge über alle Knoten (Aktivitäten und Zwischenereignisse) gebildet, wobei die Mächtigkeit der Elemente der Potenzmenge beschränkt werden kann und die leere Menge explizit aus der Potenzmenge ausgeschlossen wird. Für jedes Element der Potenzmenge, also für jede Menge an Knoten eines Modells, werden die aggregierten Merkmale bestimmt. Diese sind:

- Nummer: Eine laufende Nummer zur Indizierung der Knotenmenge
- Knotenmenge: Eine Menge der IDs der einzelnen Knoten.
- Länge: Die Mächtigkeit der Knotenmenge.
- Wiederholbarkeit: Der gemäß Definition 4.16 gemittelte Wert der einzelnen Wiederholbarkeiten.
- Zentroid: Die gemäß Definition 4.13 errechnete mittlere Position der Knotenmenge (Mittelung über die einzelnen Positionen).
- Optionalität: Der gemäß Definition 4.19 gemittelte Wert der einzelnen Optionalitäten.
- Gemeinsame Agentenmenge: Eine Menge der Agenten der einzelnen Knoten. Diese Menge wird, wie in Gleichung (4.2) angegeben, als Schnittmenge der einzelnen Agenten(mengen) gebildet.
- Gemeinsame Beschreibung: Die Konkatenation der Beschreibungen der einzelnen Knoten. Die Konkatention erfolgt zur Zeit in der Reihenfolge, in der die einzelnen Knoten in der CSV-Datei auftauchen.
- Strafterm Position: Der gemäß Definition 4.21 berechnete Strafterm zur Position der Knotenmenge; die beiden einzelnen Strafterme sind bereits addiert.

- Strafterm Optionalität: Der gemäß Definition 4.21 berechnete Strafterm zur Optionalität der Knotenmenge; die beiden einzelnen Strafterme sind bereits addiert.
- Strafterm Wiederholbarkeit: Der gemäß Definition 4.21 berechnete Strafterm zur Wiederholbarkeit der Knotenmenge; die beiden einzelnen Strafterme sind bereits addiert.
- Gemeinsame Datenmenge: Die Menge aller den einzelnen Knoten der Knotenmenge zugewiesenen Datenobjekte.

5.2.2.2 Ausschlusskriterium für Knotenmengen

Um die Anzahl der durch Potenzmengenbildung entstehenden Knotenmengen zu reduzieren und die kombinatorische Explosion etwas einzudämmen, kann eine Regelung ähnlich der der häufigen Itemsets im Apriori-Algorithmus (siehe unten) aufgestellt werden, nach der bestimmte Knotenmengen aus der Potenzmenge ausgeschlossen werden. Für eine Menge aus zwei Knoten betrachte

- die gemeinsame Agentenmenge,
- die gemittelte Wiederholbarkeit und
- die gemittelte Optionalität.

Ist die gemeinsame Agentenmenge leer, d. h., die beiden Knoten haben keinen gemeinsamen Agenten, und haben die gemittelte Wiederholbarkeit und die gemittelte Optionalität beide jeweils den Wert $0,5$, d. h., die zwei Knoten unterscheiden sich sowohl in ihrer Wiederholbarkeit als auch in ihrer Optionalität, dann nimm diese Menge an Knoten aus der Potenzmenge heraus. Es wird hier angenommen, dass es nicht sinnvoll ist, Knoten, die keinen gemeinsamen Agenten haben und sich in Wiederholbarkeit und Optionalität unterscheiden, in einer Menge zusammenzufassen. Das Ausschlusskriterium kann verschärft werden, wenn beispielsweise verlangt wird, dass nur zwei der drei genannten Bedingungen erfüllt sein müssen, um die Knotenmenge aus der Potenzmenge zu entfernen. Es können auch beliebige andere Bedingungen gestellt werden, die einzelne Perspektivenähnlichkeiten miteinander kombinieren, um unzureichend homogene Knotenmengen und somit komplette Abbildungen von vornherein auszuschließen.

Analog zum Finden häufiger Itemsets im Zuge des Apriori-Algorithmus, der beispielsweise zur Assoziationsanalyse (Warenkorbanalyse) eingesetzt wird (Goethals, 2009), werden zusätzlich zu den zweielementigen Knotenmengen weitere Mengen aus der Potenzmenge ausgeschlossen. Alle Obermengen der bereits ausgeschlossenen zweielementigen Knotenmengen werden ebenfalls aus der Potenzmenge entfernt (Monotonie-Eigenschaft, *downward closure property*), denn es gilt: Ist der Schnitt zweier Mengen A und B leer, $A \cap B = \emptyset$, so ist auch der Schnitt von A und B und beliebigen weiteren Mengen C_i leer: $A \cap B \bigcap_i C_i = \emptyset$. Für die mittlere Wiederholbarkeit und Optionalität gilt ebenfalls, dass Obermengen von unheitlichen, zweielementigen Mengen uneinheitlich bleiben, also Werte aus $(0, 1)$ annehmen.

5.2.2.3 Aufstellen der Ähnlichkeitsmatrix

Für die übriggebliebenden Elemente der beiden Potenzmengen werden nun paarweise die Ähnlichkeiten berechnet. Jede (übrig gebliebene) Knotenmenge des ersten Prozessmodells wird mit jeder (übrig gebliebenen) Knotenmenge des zweiten Prozessmodells abgeglichen.

Die Berechnung lässt sich am einfachsten in Matrixdarstellung veranschaulichen. Tabelle 5.3 zeigt den schematischen Aufbau einer solchen Matrix anhand von je drei Aktivitäten pro Modell. Die Einträge in den Zellen sind vektorwertig und enthalten folgende Ähnlichkeiten bzw. Strafwerte:

$$(BSim, ASim, VSim^{\pi}, pen^{\pi}, VSim^{o}, pen^{o}, VSim^{\rho}, pen^{\rho}, DSim) \tag{5.1}$$

Die Strafterme $pen^{\xi}(p_1, p_2) = pen^{\xi}(p_1) + pen^{\xi}(p_2)$, $\xi \in \{\pi, o, \rho\}$, sind separat angegeben, d. h. noch nicht mit dem zugehörigen Ähnlichkeitswert verrechnet, um die Verhaltensähnlichkeit nach Bedarf sowohl ohne als auch mit Straftermen berechnen zu können. Für $BSim$ wird die Levenshtein-Ähnlichkeit (Abschnitt 2.2.2.1) auf die konkatenierten Beschriftugnen angewendet.

Tabelle 5.3: Schematische Darstellung einer Ähnlichkeitsmatrix, wobei „Werte“ einen vektorwertigen Eintrag gemäß Gleichung (5.1) mit Ähnlichkeitswerten bezeichnet.

	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	$\{A,B,C\}$
$\{A'\}$	Werte	Werte	Werte	Werte	Werte	Werte	Werte
$\{B'\}$	Werte	Werte	$\cdots$				
$\{C'\}$	Werte	$\ddots$					
$\{A',B'\}$	$\vdots$						
$\{A',C'\}$							
$\{B',C'\}$							
$\{A',B',C'\}$							

Für die Weiterverarbeitung der Einzelähnlichkeiten im Zuge der ganzzahligen, linearen Optimierung, wird die Matrix nicht als Matrix übergeben, sondern als Liste aus Vektoren, deren ersten beide Einträge die laufenden Nummern der Potenzmengenelemente sind, jeweils bei 1 beginnend, gefolgt von den Einträgen der Matrix, die alle jeweils Werte zwischen 0 und 1 annehmen. Ein Ausschnitt aus einer solchen Liste ist in Listing 5.1 gegeben, wobei das Listing zeilenweise zu lesen ist und jede Zeile einen Matrixeintrag darstellt. Die ersten beiden Einträge $G1_i$ und $G2_j$ bezeichnen die Indizes der Potenzmengenelemente der beiden verglichenen Prozessmodelle G_1 und G_2; die verschiedenen Einträge sind mit einem Doppelpunkt voneinander getrennt:

$$G1_i : G2_j : BSim, ASim : VSim^{\pi} : pen^{\pi} : VSim^{o} : pen^{o} : VSim^{\rho} : pen^{\rho} : DSim$$

An das Optimierungsprogramm wird außerdem eine Zuordnungsliste im CSV-Format von laufender Nummer und Knotenmenge (IDs der einzelnen Knoten) übergeben, damit die Randbedingung, dass jeder Knoten in der Abbildung genau einmal auftauchen muss, überprüft werden kann.

Die beiden Python-Programme finden sich im ergänzenden Material zur Dissertation auf dem EPub-Server der Universität Bayreuth.[6]

[6] `epub.uni-bayreuth.de`, wird in der begutachteten Version vervollständigt. Das Python-Programm

Listing 5.1: Ähnlichkeitsmatrix in Listendarstellung (paarweise Ähnlichkeiten der Modelle „Köln“ und „Frankfurt“ aus Abschnitt 5.3).

```
1:1:0.16000000000000003:0.0:0.9761904761904763:0.0:1.0:0.0:1.0:0.0:1.0
2:1:0.19999999999999996:0.0:0.7142857142857143:0.0:1.0:0.0:1.0:0.0:1.0
3:1:0.28:0.0:0.6428571428571429:0.0:0.0:0.0:1.0:0.0:1.0
4:1:0.19999999999999996:0.0:0.9761904761904763:0.0:1.0:0.0:1.0:0.0:1.0
5:1:0.21568627450980393:1.0:0.942857142857143:0.0:0.0:0.0:1.0:0.0:1.0
6:1:0.23076923076923073:1.0:0.8428571428571429:0.0:1.0:0.0:1.0:0.0:1.0
7:1:0.6799999999999999:1.0:0.9571428571428571:0.0:1.0:0.0:1.0:0.0:1.0
8:1:0.16000000000000003:0.0:0.7142857142857143:0.0:1.0:0.0:1.0:0.0:1.0
9:1:0.16000000000000003:1.0:0.7428571428571429:0.0:0.0:0.0:1.0:0.0:1.0
10:1:0.16000000000000003:0.0:0.8095238095238095:0.0:0.0:0.0:1.0:0.0:1.0
11:1:1.0:1.0:0.9571428571428571:0.0:1.0:0.0:1.0:0.0:1.0
12:1:0.19999999999999996:1.0:0.8428571428571429:0.0:1.0:0.0:1.0:0.0:1.0
13:1:0.28:0.0:0.4761904761904763:0.0:0.0:0.0:1.0:0.0:1.0
14:1:0.16000000000000003:1.0:0.6428571428571429:0.0:0.0:0.0:1.0:0.0:1.0
15:1:0.19999999999999996:0.0:0.30952380952380953:0.0:0.0:0.0:1.0:0.0:1.0
1:2:0.125:0.0:0.7916666666666666:0.0:1.0:0.0:1.0:0.0:1.0
2:2:0.125:0.0:0.9464285714285714:0.0:1.0:0.0:1.0:0.0:1.0
3:2:0.16666666666666663:0.0:0.875:0.0:0.0:0.0:1.0:0.0:1.0
```

5.2.3 Finden einer besten Abbildung

Die Maximierung des Ähnlichkeitswerts wird mithilfe ganzzahliger, linearer Optimierung (*integer linear programming*, ILP) erreicht. Dabei muss die zu optimierende Zielfunktion linear sein und die Randbedingungen, eine Menge an Gleichungen und Ungleichungen, ebenso. Die vorkommenden unabhängigen Variablen dürfen nur ganzzahlige Werte annehmen. Im Fall des Findens einer besten Abbildung zwischen zwei Prozessmodellen und in Fortführung der ersten beiden Implementierungsschritte wird die Zielfunktion als Maximum des gewichteten Mittels festgelegt:

$$\begin{aligned}\max_M \frac{1}{8}\Big(& ASim_M(G_1,G_2) + BSim_M(G_1,G_2) + DSim_M(G_1,G_2) \\ & + penVSim^{\pi}_M(G_1,G_2) + penVSim^{\rho}_M(G_1,G_2) + penVSim^{o}_M(G_1,G_2) \\ & + NSim_M(G_1,G_2) + ESim_M(G_1,G_2)\Big)\end{aligned}$$

Die Gewichte w_i aus Definition 4.33 sind zunächst gleichverteilt gewählt ($w_i = 1/8\ \forall i$), sodass jedes Einzelähnlichkeitsmaß zu einem gleichen Anteil in die Zielfunktion eingeht. Gemäß des modularen Vorgehens, das in Abschnitt 4.4.4 beschrieben ist, kann die Gewichtung aber auch unabhängig von den gewählten Perspektivenähnlichkeiten angepasst werden.

Bei der Maximierung der Zielfunktion müssen verschiedene Nebenbedingungen gelten, unter anderem muss sichergestellt sein, dass nur gültige Abbildungen gewählt werden, d. h., eine Aktivitätenmenge aus Modell G_1 muss einer Aktivitätenmenge aus G_2 zugeordnet sein, wobei eine der beiden Aktivitätenmengen jeweils leer sein darf, und alle Aktivitäten in diesen Aktivtätenmengen dürfen in derselben Abbildung nicht noch einmal verwendet werden.

Die Eingabedateien für das ILP zur Ähnlichkeitsbestimmung der Prozessmodelle G_1 und G_2 sind die Knotenmengen N_1 und N_2, die Kantenmengen $E_1 \subseteq N_1 \times N_1$ und $E_2 \subseteq N_2 \times N_2$,

`ModelMaker.py` bewirkt das Einlesen der Modelle und das Auslesen bzw. Berechnen der Aktivitätseigenschaften wie relative Position oder Optionalität. Das Python-Programm `ModelMatcher.py` bildet die Potenzmenge der Aktivitäten bzw. alle Teilmengen bis zu einer festgelegten Größe, indiziert diese und berechnet für je zwei Mengen die verschiedenen Perspektivenähnlichkeiten.

zwei Listen mit den Indizierungen der Potenzmengenelemente beider Knotenmengen sowie eine Liste mit den Ähnlichkeiten je zweier Potenzmengenelemente. Es bezeichne

- $\mathcal{P}_i := \mathcal{P}(N_i) = 2^{N_i}$ die Potenzmenge von N_i, $i = 1, 2$ und
- $\mathcal{I}_i$ die Indexmenge zur Potenzmenge $\mathcal{P}_i$, $i = 1, 2$.

In der Ähnlichkeitsliste stehen, wie in Listing 5.1 bereits gesehen, Einträge der Form

- $i, j, ASim(\mathcal{P}_1(i), \mathcal{P}_2(j)), BSim(\mathcal{P}_1(i), \mathcal{P}_2(j)), \ldots$ mit $i \in \mathcal{I}_1, j \in \mathcal{I}_2$.

Um die Nebenbedingungen aufzustellen, werden zunächst die zwei folgenden Hilfsvariablen gebildet. Dass diese Variablen genau die Eigenschaft haben, die ihnen jeweils zugedacht ist, wird später mittels Nebenbedingungen sichergestellt.

- Die Variable $x_{k,i,j}$ ist dann 1, wenn die Abbildung M Mächtigkeit k hat und Aktivitätenmenge $\mathcal{P}_1(i)$ auf Aktivitätenmenge $\mathcal{P}_2(j)$ abgebildet wird:

$$x_{k,i,j} = \begin{cases} 1, & \text{falls } M(\mathcal{P}_1(i)) = \mathcal{P}_2(j) \wedge |M| = k \\ 0, & \text{sonst,} \end{cases}$$

  wobei $k \in K_M := \{1, \ldots, \min\{|N_1|, |N_2|\}+1\}$, $i \in \mathcal{I}_1, j \in \mathcal{I}_2$. Die maximale Mächtigkeit, die die Abbildung M haben kann, ist der Wert des größten Elements aus K_M. Es kann maximal so viele Abbildungstupel geben, wie das kleinere Prozessmodell Aktivitäten hat plus 1 für die leere Menge: $\max(K_M) = \min\{|N_1|, |N_2|\} + 1$.

- Die Variable v_k^M ist dann 1, wenn Abbildung M Mächtigkeit k hat:

$$v_k^M = \begin{cases} 1, & \text{falls } |M| = k \\ 0, & \text{sonst,} \end{cases}$$

  $\forall k \in K_M$.

5.2.3.1 Nebenbedingungen für die Perspektivenähnlichkeiten und den Anteil der abgebildeten Knoten

Mit den festgelegten Hilfsvariablen lassen sich die Perspektivenähnlichkeiten der Modelle G_1 und G_2 in der Zielfunktion wie folgt schreiben:

- $ASim_M(G_1, G_2) = \sum_{k \in K_M} \sum_{\substack{i \in \mathcal{I}_1, \\ |\mathcal{P}_0(i)|>0}} \sum_{\substack{j \in \mathcal{I}_2, \\ |\mathcal{P}_2(j)|>0}} x_{k,i,j} \cdot \frac{1}{k} \cdot ASim(\mathcal{P}_1(i), \mathcal{P}_2(j))$

Der Term $1/k$ ist die Mittelung über die Anzahl der Abbildungstupel. Die Variable $x_{k,i,j}$ ist nur für ein k gleich 1, weswegen für alle anderen $k \in K_M$ die Summanden wegen der Multiplikation mit $x_{k,i,j}$ gleich 0 sind.

- $BSim_M(G_1, G_2) = \sum_{k \in K_M} \sum_{\substack{i \in \mathcal{I}_1, \\ |\mathcal{P}_0(i)|>0}} \sum_{\substack{j \in \mathcal{I}_2, \\ |\mathcal{P}_2(j)|>0}} x_{k,i,j} \cdot \frac{1}{k} \cdot BSim(\mathcal{P}_1(i), \mathcal{P}_2(j))$

- $DSim_M(G_1, G_2) = \sum_{k \in K_M} \sum_{\substack{i \in \mathcal{I}_1, \\ |\mathcal{P}_0(i)|>0}} \sum_{\substack{j \in \mathcal{I}_2, \\ |\mathcal{P}_2(j)|>0}} x_{k,i,j} \cdot \frac{1}{k} \cdot DSim(\mathcal{P}_1(i), \mathcal{P}_2(j))$

- $penVSim_M^{\xi}(G_1, G_2)$
  $= \sum_{k \in K_M} \sum_{\substack{i \in \mathcal{I}_1, \\ |\mathcal{P}_0(i)|>0}} \sum_{\substack{j \in \mathcal{I}_2, \\ |\mathcal{P}_2(j)|>0}} x_{k,i,j} \cdot \frac{1}{k} \cdot (VSim(\mathcal{P}_1(i), \mathcal{P}_2(j)) - pen^{\xi}(\mathcal{P}_1(i), \mathcal{P}_2(j)))$

Es ist $\xi \in \{\pi, \rho, o\}$. Der Term $pen^{\xi}(\cdot, \cdot)$ ist die Summe der einzelnen Strafterme: $pen^{\xi}(\mathcal{P}_1(i), \mathcal{P}_2(j)) = pen^{\xi}(\mathcal{P}_1(i)) + pen^{\xi}(\mathcal{P}_2(j))$. Dieser wird in der Ähnlichkeitsmatrix aus Abschnitt 5.2.2.3 bereits aufsummiert übergeben.

Der Anteil der abgebildeten Knoten $NSim_M$ wird wie in Definition 4.34 über „1 − Anteil gelöschter Knoten“ bestimmt. Es wird also die Mächtigkeit der Aktivitätenmenge benötigt, die auf die leere Menge abgebildet wird bzw. die das Bild der leeren Menge ist, sofern es diese Mengen gibt. Wird die Menge mit Index i aus dem ersten Prozessmodell auf die leere Menge abgebildet, so ist $x_{k,i,0}$ für ein bestimmtes k gleich 1. Ist die Menge mit Index j das Bild der leeren Menge, so ist für ein bestimmtes k die Variable $x_{k,0,j}$ gleich 1. Der Anteil der abgebildeten Knoten ist

- $NSim_M(G_1, G_2) = 1 - \frac{1}{|N_1|+|N_2|} \left(\sum_{k \in K_M} \left(\sum_{i \in \mathcal{I}_1} x_{k,i,0} \cdot |\mathcal{P}_1(i)| + \sum_{j \in \mathcal{I}_2} x_{k,0,j} \cdot |\mathcal{P}_2(j)| \right) \right)$

Für die Abbildung M gelten die folgenden Nebenbedingungen. Jede Aktivität $n \in N_1$ muss für ein festgelegtes k in genau einer Aktivitätenmenge, d. h. in einem Element der Potenzmenge von N_1, enthalten sein. Gleiches gilt für jede Aktivität $m \in N_2$. Außerdem ist für dieses k eine Aktivitätenmenge aus G_1 genau einer Aktivitätenmenge aus G_2 zugeordnet.

- $\sum_{k \in K_M} \sum_{\substack{i \in \mathcal{I}_1, \\ |\mathcal{P}_1(i) \cap \{n\}|>0}} \sum_{j \in \mathcal{I}_2} x_{k,i,j} = 1, \forall n \in N_1$ (5.2)

- $\sum_{k \in K_M} \sum_{i \in \mathcal{I}_1} \sum_{\substack{j \in \mathcal{I}_2, \\ |\mathcal{P}_2(j) \cap \{m\}|>0}} x_{k,i,j} = 1, \forall m \in N_2$ (5.3)

Die leere Menge darf nicht auf die leere Menge abgebildet werden, die in beiden Potenzmengen den Index 0 hat. Außerdem ist $x_{k,i,j}$ eine Binärvariable.

- $x_{k,0,0} = 0, \forall k \in K_M$
- $x_{k,i,j} \in \{0, 1\}, \forall k \in K_M, i \in \mathcal{I}_1, j \in \mathcal{I}_2$

Dass die Abbildung M genau Mächtigkeit k hat, also dass v_k^M nur für genau ein k gleich 1 ist, wird sichergestellt mittels

- $\sum_{k \in K_M} v_k^M = 1$, wobei
- $v_k^M \in \{0, 1\}, \forall k \in K_M$.

Damit die Variable $x_{k,i,j}$ genau die Bedeutung hat, die ihr zugedacht ist, also dass sie genau dann 1 ist, wenn die Abbildung k Elemente enthält und die Aktivitätenmenge mit Index i auf die mit Index j abgebildet wird, muss zusammen mit den bisherigen Gleichungen (insbesondere zusammen mit den Gleichungen (5.2) und (5.3)) gelten:

- $k \cdot v_k^M = \sum_{i \in \mathcal{I}_1} \sum_{j \in \mathcal{I}_2} x_{k,i,j}, \forall k \in K_M$

Damit sind die beiden Variablen v und x gekoppelt und somit alle Randbedingungen für eine korrekte Abbildung M festgelegt. In der Menge der zulässigen Lösungen liegen alle Abbildungen, die die aufgestellten Kriterien, d. h. insbesondere die Gleichungen und Ungleichungen, erfüllen. Optimiert wird der maximale Ähnlichkeitswert, der durch diese Abbildungen gemäß der Zielfunktion berechnet wird. Die Berechnungsvorschrift des Anteils der abgebildeten Kanten, $ESim_M$, und dessen Randbedingungen sind noch festzulegen.

5.2.3.2 Nebenbedingungen für Kantenanteil

Es bezeichne $e^i_{n,m}$ die Kante aus E_i von n nach m, also

$$e^i_{n,m} = 1 \;\Leftrightarrow\; (n,m) \in E_i, i = 1,2.$$

Die Hilfsfunktion $hasArc(j_1, j_2)$ legt fest, ob in den gegebenen Prozessmodellen zwischen zwei Aktivitätenmengen j_1 und j_2 eine Kante besteht. Eine Kante zwischen zwei Aktivitätenmengen existiert genau dann, wenn zwischen einer Aktivität aus der Aktivitätenmenge j_1 und einer aus j_2 eine Kante besteht. Es werden somit alle Kanten, die in $\bigcup_M \tilde{E}_{M,i}$, der Menge aller abstrahierten Kanten (siehe Definition 4.37) für alle möglichen Abbildungen M, liegen, mit einer 1 markiert. Diese sind, wegen der Vereinigung über M, unabhängig von der zugrunde liegenden Abbildung und benötigen auch noch keine Information über die jeweiligen Partitionen. Der Funktionswert $hasArc(\cdot,\cdot)$ kann für zwei beliebige Elemente der Potentmenge der Aktivitäten bestimmt werden. Die Listen aller Kanten werden für jedes Prozessmodell separat dem ILP übergeben.

$$hasArc_i(j_1, j_2) = \begin{cases} 1, & \text{falls} \sum\limits_{\substack{n \in \mathcal{P}_i(j_1),\\ m \in \mathcal{P}_i(j_2)}} e^i_{n,m} \geq 1 \\ 0, & \text{sonst,} \end{cases}$$

$i \in \{1,2\}, \forall j_1, j_2 \in \mathcal{I}_i$.

Der Anteil der abgebildeten Kanten wird für die abstrahierten Kanten, wie in Definition 4.39 gegeben, also für Kanten zwischen Aktivitätenmengen, berechnet und nicht für Kanten zwischen einzelnen Aktivitäten. Eine Kante, die abgebildet wird, also in $Sube''_M$ (siehe Definition 4.38) liegt, wird im folgenden kurz als gute Kante bezeichnet.

- Es wird eine weitere Hilfsvariable benötigt, die die guten Kanten als solche kennzeichnet. Diese Hilfsvariable hat die folgende Bedeutung, nämlich

  $$g^i_{\ell,j_1,j_2} = \begin{cases} 1, & \text{falls Kante } (\mathcal{P}_i(j_1), \mathcal{P}_i(j_2)) \text{ in } G_i \text{ eine gute Kante ist} \\ & \text{und insgesamt } \ell \text{ Kanten existieren} \\ 0, & \text{sonst} \end{cases}$$

  $\forall i \in \{1,2\}, \forall \ell \in K_e, \forall j_1, j_2 \in \mathcal{I}_i$. Hierbei ist $K_e := \{1, \ldots, |E_1| + |E_2|\}$.

- Eine zweite Hilfsvariable legt fest, dass, wenn es überhaupt zählende Kanten gibt, es in beiden Modellen zusammen genau ℓ Kanten gibt, die zählen, also dass die Mächtigkeit von $\tilde{E}_{M,1} + \tilde{E}_{M,2}$, der Mengen der abstrahierten Kanten (siehe Definition 4.37), gleich ℓ ist:

  $$v^e_\ell = \begin{cases} 1, & \text{falls insgesamt } \ell \text{ Kanten zählen} \\ 0, & \text{sonst,} \end{cases}$$

  $\forall \ell \in K_e$.

- Eine dritte Hilfsvariable deckt den Fall ab, dass es keine zählenden Kanten gibt. Dieser ist auch in Definition 4.39 separat ausgenommen:

$$\eta = \begin{cases} 0, & \text{wenn } \ell \text{ Kanten zählen } (\ell \in K_e) \\ 1, & \text{wenn es keine zählenden Kanten gibt} \end{cases}$$

- Eine vierte Hilfsvariable kennzeichnet die zählenden Kanten als solche:

$$c^i_{j_1,j_2} = \begin{cases} 1, & \text{falls Kante } (\mathcal{P}_i(j_1), \mathcal{P}_i(j_2)) \text{ in } G_i \text{ zählt} \\ 0, & \text{sonst} \end{cases}$$

Die Variablen v^e_ℓ und $c^i_{j_1,j_2}$ werden gekoppelt um sicherzustellen, dass sie ihre angedachte Bedeutung erfüllen. Falls es zählende Kanten gibt, so gibt es genau ℓ zählende (abstrahierte) Kanten:

- $\sum\limits_{\ell \in K_e} v^e_\ell \leq 1$, wobei
- $v^e_\ell \in \{0,1\}$, $\forall \ell \in K_e$

Diese ℓ zählenden Kanten sind Kanten zwischen Aktivitätenmengen der jeweiligen Prozessmodelle:

- $\sum\limits_{\ell \in K_e} \ell \cdot v^e_\ell = \sum\limits_{i_1,i_2 \in \mathcal{I}_1 \times \mathcal{I}_1} c^1_{i_1,i_2} + \sum\limits_{j_1,j_2 \in \mathcal{I}_2 \times \mathcal{I}_2} c^2_{j_1,j_2}$

Damit $c^i_{j_1,j_2}$ tatsächlich die gewünschte Bedeutung hat, die zählenden Kanten als solche zu kennzeichnen, werden folgende Bedingungen getrennt für beide Prozessmodelle aufgestellt:

- $c^1_{j_1,j_2} \leq hasArc_1(j_1,j_2)$, $\forall j_1, j_2 \in \mathcal{I}_1$

Eine zählende Kante kann es nur dann geben, wenn es überhaupt eine Kante zwischen Aktivitäten der untersuchten Aktivitätenmengen gibt. Zählende Kanten hängen von der Abbildung bzw. den Partitionen der Aktivitäten der untersuchten Prozessmodelle ab. Sie stellen eine Auswahl der Kanten dar, die von der Funktion *hasArc* als existent gekennzeichnet wurden.

- $\left(\left(\sum\limits_{j_1 \in \mathcal{I}_2} x_{k,i_1,j_1} + \sum\limits_{j_2 \in \mathcal{I}_2} x_{k,i_2,j_2}\right) - 1\right) \cdot hasArc_1(i_1,i_2) \leq c^1_{i_1,i_2}$, $\forall k \in K_M, \forall i_1, i_2 \in \mathcal{I}_1$
- $\sum_{j_1 \in \mathcal{I}_2} x_{k,i_1,j_1} + \sum_{j_2 \in \mathcal{I}_2} x_{k,i_2,j_2} \geq 2 \cdot c^1_{i_1,i_2}$, $\forall k \in K_M, \forall i_1, i_2 \in \mathcal{I}_1$

Wenn es eine Kante zwischen i_1 und i_2 gibt und i_1 auf j_1 und i_2 auf j_2 abgebildet wird, dann ist zwischen i_1 und i_2 eine zählende Kante in G_1. Analoge Ungleichungen werden für zählende Kanten in G_2 aufgestellt:

- $c^2_{j_1,j_2} \leq hasArc_2(j_1,j_2)$, $\forall j_1, j_2 \in \mathcal{I}_2$
- $\left(\left(\sum\limits_{i_1 \in \mathcal{I}_1} x_{k,i_1,j_1} + \sum\limits_{i_2 \in \mathcal{I}_1} x_{k,i_2,j_2}\right) - 1\right) \cdot hasArc_2(i_1,i_2) \leq c^2_{i_1,i_2}$, $\forall k \in K_M, \forall j_1, j_2 \in \mathcal{I}_2$

- $\sum_{i_1 \in \mathcal{I}_1} x_{k,i_1,j_1} + \sum_{i_2 \in \mathcal{I}_1} x_{k,i_2,j_2} \geq 2 \cdot c^2_{i_1,i_2}, \forall k \in K_M, \forall j_1, j_2 \in \mathcal{I}_2$

Außerdem ist jedes $c^i_{j_1,j_2}$ entweder 0 oder 1:

- $c^i_{j_1,j_2} \in \{0,1\}\ \forall i \in \{1,2\}, \forall j_1, j_2 \in \mathcal{I}_i$

Aus der Menge der zählenden (abstrahierten) Kanten, werden nun die guten Kanten mit folgenden Nebenbedingungen ausgewählt. Eine Kante kann nur dann gut sein, wenn sie überhaupt zählt. Außerdem kann es, wenn ℓ Kanten zählen, nur maximal eine gute Kante zwischen j_1 und j_2 geben.

- $g^i_{\ell,j_1,j_2} \leq c^i_{j_1,j_2}, \forall \ell \in K_e, \forall j_1, j_2 \in \mathcal{I}_i, \forall i \in \{1,2\}$
- $v^e_\ell \geq g^i_{\ell,j_1,j_2}, \forall \ell \in K_e, \forall j_1, j_2 \in \mathcal{I}_i, \forall i \in \{1,2\}$

Eine zählende Kante zwischen i_1 und i_2 ist dann eine gute Kante, wenn i_1 auf j_1 und i_2 auf j_2 abgebildet wird und auch zwischen j_1 und j_2 eine Kante existiert. Dies gilt in beiden Prozessmodellen.

- $\sum_{\ell \in K_e} 2g^1_{\ell,i_1,i_2} \leq \left(\sum_{k \in K_M} \left(\sum_{j_1 \in \mathcal{I}_2} x_{k,i_1,j_1} + \sum_{j_2 \in \mathcal{I}_2} x_{k,i_2,j_2} \right) \right) \cdot hasArc_2(j_1, j_2),\ \forall i_1, i_2 \in \mathcal{I}_1$
- $\sum_{\ell \in K_e} 2g^2_{\ell,j_1,j_2} \leq \left(\sum_{k \in K_M} \left(\sum_{i_1 \in \mathcal{I}_1} x_{k,i_1,j_1} + \sum_{i_2 \in \mathcal{I}_1} x_{k,i_2,j_2} \right) \right) \cdot hasArc_1(i_1, i_2),\ \forall j_1, j_2 \in \mathcal{I}_2$
- $g^i_{\ell,j_1,j_2} \in \{0,1\},\ \forall \ell \in K_e, \forall j_1, j_2 \in \mathcal{I}_i$

Es fehlt nun noch die Möglichkeit, dass es gar keine zählenden Kanten gibt, also dass $\sum_{\ell \in K_e} v^e_\ell = 0$. Dann sind auch alle $c^i_{j_1,j_2} = 0$ und somit auch alle $g^i_{\ell,j_1,j_2} = 0$. Die Variable η soll dann und nur genau dann den Wert 1 erhalten.

- $1 - \sum_{\ell \in K_e} v^e_\ell \geq \eta$

Damit ist $ESim$, das sich als Summe der guten Kanten durch die Summe der zählenden Kanten schreiben lässt,

- $ESim_M(G_1, G_2) = \sum_{\ell \in K_e} \frac{1}{\ell} \left(\sum_{i_1,i_2 \in \mathcal{I}_1} g^1_{\ell,i_1,i_2} + \sum_{j_1,j_2 \in \mathcal{I}_2} g^2_{\ell,j_1,j_2} \right) + \eta.$

Dadurch dass es in G_1 immer genau so viele gute Kanten wie in G_2 gibt, da gute Kanten genau die abgebildeten Kanten sind, ließen sich die Nebenbedingungen auch etwas vereinfachen. Es genügt, die guten Kanten für ein Modell zu bestimmen, denn es wird zur Berechnung von $ESim$ nur die Anzahl der guten Kanten, nicht deren genaue Lage, benötigt. Das Addieren von η bewirkt, dass $ESim_M(G_1, G_2)$ im Fall, dass es keine zählenden Kanten gibt, den Wert 1 annimmt. Der erste Teil des Terms ist dann 0, da, wie oben geschrieben, keine guten Kanten existieren. Da in der Formel für $ESim$ durch ℓ geteilt wird, ist es nicht möglich, in die Menge K_e die Null einfach mit einzuschließen.

Insgesamt sind die angegebenen Nebenbedingungen hinreichend, um das gewünschte Verhalten der Abbildung von Knoten und Kanten zu bewirken, aber womöglich nicht alle notwendig. Als Proof of Concept-Implementierung reicht dies jedoch aus. Der ZIMPL-Code, der

die Zielfunktion und die eben aufgestellten Nebenbedingungen beinhaltet, ist in Anhang A.5 gezeigt.[7] Hier ist statt G_1 und G_2 die Indizierung G_0 und G_1 verwendet.

Anhand der Definitionen in Abschnitt 4.4.3 und des Implementierungsaufwands im ILP lässt sich leicht sehen, dass vor allem der Kantenanteil und dessen Berechnung mit einem hohen Aufwand verbunden sind. Die Berechnung des Kantenanteils wurde aus der Literatur (siehe z. B. Dijkman et al., 2009b) übernommen. Dadurch dass beim multiperspektivischen Abgleich das Verhalten explizit als Teilähnlichkeitswert in die Zielfunktion eingeht und die Kanten eines Prozessmodells zumindest zu einem gewissen Maß das Verhalten beeinflussen bzw. einen Teil des Verhaltens widerspiegeln, stellt sich die Frage, ob $ESim$ in diesem Fall nicht vernachlässigbar ist. Darüber hinaus ist es bei einem Abgleich von deklarativen Prozessmodellen nicht möglich, einen Kantenanteil zu bestimmen. Hier ist $ESim$ also in der Zielfunktion nicht enthalten. Es kann in weiterführenden Arbeiten überprüft werden, wie groß der Einfluss des Kantenanteils auf das Abgleichsergebnis tatsächlich ist. In der vorliegenden Arbeit wird dieser Frage nicht konkret nachgegangen, da es für die Auswertung in Abschnitt 5.3 nicht notwendig ist.

Abgesehen von dem eben erwähnten Kantenanteil ist die Formulierung der Optimierung als ILP nicht allzu kompliziert, ermöglicht es aber, eine optimale Lösung des Optimierungsproblems zu finden, ohne einen kompletten Brute-Force-Ansatz benutzen zu müssen. Auch für einen Greedy-Ansatz, der jedoch nicht unbedingt eine optimale Lösung findet, müssen paarweise Ähnlichkeiten bestimmt werden, sodass die Vorarbeit in den Implementierungsschritten aus Abschnitt 5.2.1 und 5.2.2 auch für einen solchen notwendig ist. Ein Greedy-Algorithmus lässt zwar im Gegensatz zum ILP einen Laufzeitvorteil erwarten, doch wie am Ende von Abschnitt 4.4.4 geschrieben, kann es notwendig sein, einen Greedy-Algorithmus mehrfach durchlaufen zu lassen, wenn die zunächst gefundene Abbildung nicht gut genug ist. Außerdem ist es gerade bei Ähnlichkeitsabgleichen, die Konformitäts-, Verständlichkeits- oder Evaluationszwecke erfüllen sollen (Abschnitte 1.1.2, 1.1.3 und 1.1.4), nicht ratsam, keine optimalen Ergebnisse zu erhalten. Je nach Anzahl der zu vergleichenden Modelle kann die Laufzeit sowieso vernachlässigt werden. Im nachfolgenden Abschnitt 5.3, in dem der eben beschriebene Abgleichsansatz angewendet wird, werden insgesamt 36 paarweise Abgleiche durchgeführt, was einem relativ kleinen Modellrepositorium entspricht.

5.3 Evaluation mit den Prozessmodellen des Process Model Matching Contests

Um die Güte des in der Arbeit entwickelten Ähnlichkeitsmaßes zu bestimmen, wird die Ähnlichkeitsbestimmung auf eine Menge an Prozessmodellen angewendet, die ihm Rahmen des Process Model Matching Contests 2013 und 2015 zur Verfügung gestellt wurden. Für diese Modelle existiert ein händisch vorgegebener Goldstandard, eine optimale Abbildung zwischen je zwei der insgesamt neun Modelle. Die Prozessmodelle stellen den Bewerbungsprozess an neun deutschen Hochschulen bzw. Universitäten dar und sind in BPMN modelliert.

Da das Python-Programm zum Einlesen der Modelle die vom Camunda Modeler verwendeten Bezeichnungen benötigt, die vorgegebenen Modelle aber andere XML-Tags verwenden, wurden die Modelle zunächst im Camunda Modeler nachmodelliert. Nach dem Einlesen der

[7] Des Weiteren ist das ZIMPL-Programm im ergänzenden Material zur Dissertation auf dem EPub-Server der Universität Bayreuth `epub.uni-bayreuth.de` (wird in der begutachteten Version vervollständigt) unter dem Namen `AbbildungMN.zpl` verfügbar.

verschiedenen Modelle stößt der verwendete Computer[8] beim zweiten Schritt, dem Bilden der Potenzmenge der Aktivitäten, an das Limit des Arbeitsspeichers. Es zeigt sich, dass die Abspeicherung als CSV-Datei für größere Modelle (ca. > 10 Aktivitäten) nicht gut geeignet ist. Ein schrittweises Schreiben der CSV-Datei könnte das Problem des Arbeitsspeicherüberlaufs zwar beheben, doch die entstehende Datei wäre trotzdem mehrere GB groß. Eine alternative Abspeicherung, zum Beispiel als HDF5-Datei[9], ein Format, das speziell für große Datenmengen ausgelegt ist, wäre vonnöten.

5.3.1 Anpassung an 1:1-Abbildung

Da der Goldstandard für die zu testenden Prozessmodelle jedoch hauptsächlich 1:1-Abbildungen vorschlägt, wird das Python-Programm dahingehend modifiziert, dass bei der Potenzmengenbildung nur Mengen mit einer maximalen Mächtigkeit von 1 zugelassen werden, also nur solche Mengen, die maximal eine Aktivität enthalten. Die Einschränkung auf 1:1-Abbildungen erfordert jedoch ebenfalls eine Modifikation des Optimierungsprogramms. Die gebildeten Aktivitätenmengen enthalten alle maximal ein Element, jedoch muss es bei einer 1:1-Abbildung trotzdem möglich sein, mehr als eine Aktivität zu löschen (nicht abzubilden). Deswegen ist es notwendig, die Bijektivitätseigenschaft der Abbildung aufzuheben, damit mehrere Aktivitätemengen auf die leere Menge angebildet werden können. Konkret wird das in den Nebenbedingungen des Optimierungsproblems wie folgt erreicht: Es wird nicht mehr als notwendig vorausgesetzt, dass jedes $n \in N_i$ in einer Aktivitätenmenge auftauchen muss. Dies wird mit folgender Bedingung ausgedrückt:

- $\sum_{k \in K_M} \sum_{\substack{i \in \mathcal{I}_1, \\ |\mathcal{P}_1(i) \cap \{n\}|>0}} \sum_{j \in \mathcal{I}_2} x_{k,i,j} \leq 1, \forall n \in N_1$
- $\sum_{k \in K_M} \sum_{i \in \mathcal{I}_1} \sum_{\substack{j \in \mathcal{I}_2, \\ |\mathcal{P}_2(j) \cap \{m\}|>0}} x_{k,i,j} \leq 1, \forall m \in N_2$

Nicht für alle (einelementigen) Elemente der „Potenzmenge“ $\mathcal{P}_j(i)$ mit $\{n\} \subseteq \mathcal{P}_j(i), n \in N_j, j = 1, 2$ muss es ein korrespondierendes Element im anderen Modell, also in der „Potenzmenge“ der Aktivitäten des anderen Modells geben. Diese beiden Bedingungen gelten statt den Gleichungen (5.2) und (5.3). Für die gelöschten Aktivitäten wird nun nicht explizit gefordert, dass sie auf die leere Menge abgebildet werden; sie können auch einfach nicht abgebildet sein. Für die Berechnung des Ähnlichkeitswert liefern beide Möglichkeiten jedoch dasselbe Ergebnis, da bei den Berechnungen der Perspektivähnlichkeiten nur Aktivitätenmengen berücksichtigt werden, die auf andere Aktivitätenmengen abgebildet werden.

Eine zweite Anpassung, die in diesem Zuge vorgenommen werden muss, ist die Berechnung von $NSim$. Da die gelöschten Knoten nicht mehr zwangsläufig diejenigen sind, die auf die leere Menge abgebildet werden, muss die Formel auf die direkte Berechnung mittels Anzahl abgebildeter Knoten geteilt durch Anzahl aller Knoten umgestellt werden. Die direkte Berechnung nutzt die Tatsache, dass wenn es genau k Elemente in einer 1:1-Abbildung gibt, genau $2k$ Aktivitäten abgebildet werden, mit Ausnahme derer, die auf die leere Menge abgebildet werden, falls solche existieren.

[8] DELL LATITUDE E6540 mit Windows 8.1 Enterprise, 64-Bit-Betriebssystem, Prozessor: Intel(R) Core i7 2,8 GHz, Arbeitsspeicher: 8 GB DDR3

[9] `https://www.hdfgroup.org/hdf5/`

- $NSim_M(G_1, G_2) = \frac{1}{|N_1|+|N_2|} \left(\sum_{k \in K_M} \left(\sum_{i \in \mathcal{I}_1} \sum_{j \in \mathcal{I}_2} 2 \cdot x_{k,i,j} - \sum_{j \in \mathcal{I}_2} x_{k,0,j} - \sum_{i \in \mathcal{I}_1} x_{k,i,0} \right) \right)$

Außerdem muss in diesem Zusammenhang auch die Berechnung des Anteils der abgebildeten Kanten angepasst werden, da gelöschte Knoten wiederum nicht alle über die Variable $x_{k,i,j}$ mit entweder $i = 0$ oder $j = 0$ abrufbar sind. Da keine Aktivitäten zu Aktivitätenmengen zusammengefasst werden, gibt es keine neutralen Kanten, d. h., die Anzahl der zählenden Kanten ist gleich der Anzahl der vorhandenen Kanten, die wiederum nicht von der Abbildung abhängt. Mit der vorhandenen Funktion *hasArc* kann die Gesamtzahl der Kanten bestimmt werden:

- $kantenzahl = \sum_{i_1,i_2 \in \mathcal{I}_1 \times \mathcal{I}_1} hasArc_1(i_1, i_2) + \sum_{i_1,i_2 \in \mathcal{I}_2 \times \mathcal{I}_2} hasArc_2(i_1, i_2)$

Die Variablen $c^i_{i,j}$ werden nicht mehr benötigt und die Markierung der guten Kanten kann unabhängig von k erfolgen. Der Anteil der abgebildeten Kanten errechnet sich aus

- $ESim_M(G_1, G_2) = \frac{1}{kantenzahl} \cdot \left(\sum_{i_1,i_2 \in \mathcal{I}_1 \times \mathcal{I}_1} g^1_{i_1,i_2} + \sum_{i_1,i_2 \in \mathcal{I}_2 \times \mathcal{I}_2} g^2_{i_1,i_2} \right)$

Das für 1:1-Abbildungen modifizierte ZIMPL-Programm ist in Anhang A.5 in Listing A.2 gegeben.[10]

5.3.2 Erste Tests

Mit diesen Anpassungen werden nun die optimalen Abbildungen für die Beispielmodelle gesucht, wobei die Gewichte der einzelnen Perspektivenähnlichkeiten in der Zielfunktion zunächst gleichverteilt sind. Abbildung 5.8 zeigt den Bewerbungsprozess der Uni Köln, Abbildung 5.9 den der Uni Frankfurt. Der Goldstandard, der bei einem Abgleich dieser beiden Modelle erreicht werden soll, ist die folgende Abbildung, wobei jeweils zuerst der Knoten aus dem Kölner Modell und die Entsprechung aus dem Frankfurter Modell genannt ist. Es sind die Aktivitätenbeschreibungen bzw. die Ereignisbeschreibungen von Zwischenereignissen (Ereignis mit doppelter Umrandung) angegeben. Für diese beiden Modelle umfasst die optimale Abbildung laut des vorgegebenen Goldstandards acht Elemente. Es werden die Aktivitäten aufeinander abgebildet, die jeweils die folgenden Beschriftungen aufweisen: „Apply online“, „Send documents by post“, „Wait for results“, „Rejected“, „Accepted“, „Check documents“, „Send letter of acceptance“ und „Send letter of rejection“.

Die Ausgabe der SCIP Optimization für diesen Abgleich ist, in gekürzter Form, in Listing 5.2 zu sehen, das im Folgenden erläutert wird. In einem ersten Schritt wird der ZIMPL-Programmcode eingelesen und in ein für den Solver geeignetes Format übersetzt (`read problem`). Dieses besteht aus einer Vielzahl an Variablen (hier: 7000 Variablen), für die eine Reihe an Bedingungen (hier: 1745 Bedingungen) gelten. Die Laufzeit zum Lösen dieses Problems betrug fast zweieinhalb Minuten (`Solving Time (sec): 84.42`), was für eine Praxislösung auf einem großen Repository etwas zu lang ist. Der maximale Ähnlichkeitswert, der für dieses Problem gefunden wird, liegt bei knapp 89 % (`objective value`). Die zugehörige Abbildung umfasst fünfzehn Elemente (`v#15` entspricht $v^M_k = 1$ für $k = |M| = 15$). Die abgebildeten

[10] Des Weiteren ist das ZIMPL-Programm im ergänzenden Material zur Dissertation auf dem EPub-Server der Universität Bayreuth `epub.uni-bayreuth.de` (wird in der begutachteten Version vervollständigt) unter dem Namen `Abbildung11.zpl` verfügbar.

Elemente sind in der Belegung der $x_{k,i,j}$ abzulesen: `x#k#i#j` entspricht $x_{k,i,j}$, wobei $k = 15$ ist und die aufgeführten i und j die jeweiligen Indizes der Potenzmengenelemente darstellen. Die Zuordnung der Indizes zu den Aktivitätenmengen, die in diesem Fall aus jeweils genau einer Aktivität bestehen, ist in den Listen, die der zweite Teil des Python-Programms zusätzlich zur Ähnlichkeitsmatrix ausgibt, abzulesen. Die Angabe `goodArcsG0#k#i#j` bzw. `goodArcsG1#k#i#j` zeigt an, zwischen welchen Aktivitätenmengen abgebildete Kanten vorhanden sind. Schließlich sind die Ähnlichkeitswerte der jeweiligen Perspektiven aufgeführt, aus denen sich, unter der vorgegebenen Gewichtung, die in Klammern hinter den jeweiligen Werten nach `obj` noch einmal angegeben sind, der Zielfunktionswert berechnet.

Listing 5.2: Ausgabe des Abgleichs Köln und Frankfurt.

```
read problem <Cologne-Frankfurt.zpl>

original problem has 7000 variables
  (6992 bin, 0 int, 0 impl, 8 cont) and 1745 constraints

SCIP Status        : problem is solved [optimal solution found]
Solving Time (sec) : 84.42
Solving Nodes      : 1 (total of 3 nodes in 3 runs)
Primal Bound       : +8.87947129786283e-01 (4 solutions)
Dual Bound         : +8.87947129786283e-01
Gap                : 0.00 %

SCIP> display solution

objective value:                    0.887947129786283
v#15                                                1   (obj:0)
x#15#1#19                                           1   (obj:0)
x#15#2#6                                            1   (obj:0)
x#15#3#4                                            1   (obj:0)
x#15#4#3                                            1   (obj:0)
x#15#5#18                                           1   (obj:0)
x#15#6#2                                            1   (obj:0)
x#15#7#17                                           1   (obj:0)
x#15#8#16                                           1   (obj:0)
x#15#9#15                                           1   (obj:0)
x#15#10#12                                          1   (obj:0)
x#15#11#1                                           1   (obj:0)
x#15#12#5                                           1   (obj:0)
x#15#13#11                                          1   (obj:0)
x#15#14#14                                          1   (obj:0)
x#15#15#9                                           1   (obj:0)
goodArcsG0#2#10                                     1   (obj:0)
goodArcsG0#3#2                                      1   (obj:0)
 ...
goodArcsG0#15#13                                    1   (obj:0)
goodArcsG1#1#3                                      1   (obj:0)
goodArcsG1#2#15                                     1   (obj:0)
 ...
goodArcsG1#23#18                                    1   (obj:0)
ASim                                                1   (obj:0.125)
BSim                                0.680090530811593   (obj:0.125)
DSim                                                1   (obj:0.125)
penVSimPi                           0.937558922558922   (obj:0.125)
penVSimOpt                          0.866666666666667   (obj:0.125)
penVSimRho                                          1   (obj:0.125)
```

```
NSim                                    0.789473684210526   (obj:0.125)
ESim                                    0.829787234042553   (obj:0.125)
```

Zusätzlich zur Optimierungsdauer, die hier bei knapp eineinhalb Minuten liegt, muss noch die Zeit des Einlesens, die für dieses Beispiel bei ca. 18 Sekunden liegt, bei der Angabe der Laufzeit berücksichtigt werden. Bei näherer Betrachtung der Prozessmodelle fällt auf, dass in diesen keine Datenobjekte verwendet werden. Diese Perspektive scheint also bei der Modellierung nicht berücksichtigt worden zu sein und somit ist es nicht sinnvoll, Ähnlichkeitswerte auf Basis dieser Perspektive, die unabhängig von der Abbildung immer volle Ähnlichkeit aufweist, mit einzubeziehen. Außerdem ist in keinem einzigen Modell eine Schleife enthalten, d. h., auf Wiederholbarkeit zu testen ist ebenfalls nicht sinnvoll. Das $DSim$ und das $VSim^{\rho}$ können also mit einer entsprechenden Anpassung der Gewichte aus der Zielfunktion in ZIMPL gestrichen werden. Die Gewichte an sich beeinflussen nicht die optimale Abbildung, sondern lediglich den absoluten Ähnlichkeitswert. Addieren sich die Gewichte zu einem Wert < 1, dann kann in keinem Fall eine Ähnlichkeit von 100 % erreicht werden. Ohne Berücksichtigung der Datenperspektive ändert sich das Ergebnis zu dem in Listing 5.3 dargestellten Optimum.

Listing 5.3: Ausgabe des Abgleichs Köln und Frankfurt ohne $DSim$ und ohne $penVSimRho$.

```
Solving Time (sec) : 69.72

objective value:                0.850596173048377
v#15                                            1   (obj:0)
x#15#1#19                                       1   (obj:0)
x#15#2#6                                        1   (obj:0)
x#15#3#4                                        1   (obj:0)
x#15#4#3                                        1   (obj:0)
x#15#5#18                                       1   (obj:0)
x#15#6#2                                        1   (obj:0)
x#15#7#17                                       1   (obj:0)
x#15#8#16                                       1   (obj:0)
x#15#9#15                                       1   (obj:0)
x#15#10#12                                      1   (obj:0)
x#15#11#1                                       1   (obj:0)
x#15#12#5                                       1   (obj:0)
x#15#13#11                                      1   (obj:0)
x#15#14#14                                      1   (obj:0)
x#15#15#9                                       1   (obj:0)

ASim                                            1   (obj:0.166666666666667)
BSim                    0.680090530811593       (obj:0.166666666666667)
penVSimPi               0.937558922558922       (obj:0.166666666666667)
penVSimOpt              0.866666666666667       (obj:0.166666666666667)
NSim                    0.789473684210526       (obj:0.166666666666667)
ESim                    0.829787234042553       (obj:0.166666666666667)
```

Dies entspricht immer noch einer großen Menge als ähnlich erkannter Aktivitäten, die im Goldstandard nicht als ähnlich deklariert sind (der Goldstandard umfasst acht ähnliche Aktivitätenpaare, nicht fünfzehn). Im Vergleich zur Optimierung mit $DSim$ und $VSim^{\rho}$ hat sich an der Abbildung an sich nichts geändert. Eine Anpassung der Gewichte, da offenbar die Bestrafung nicht abgebildeter Knoten und Kanten im Vergleich zu den übrigen Ähnlichkeitswerten sehr stark ins Gewicht fällt, kann für einen besseren Ausgleich der verschiedenen Perspektiven sorgen. Mit einer Gewichtung von 1/5 für die vier Maße $ASim$, $BSim$, $penVSim^{\pi}$ und $penVSim^{o}$ und einer Gewichtung von jeweils 1/10 für $NSim$ und $ESim$ ändert sich das Ergebnis zu der in Listing 5.4 gezeigten optimalen Abbildung.

Listing 5.4: Ausgabe des Abgleichs Köln und Frankfurt ohne $DSim$ und $VSimRho$ und mit schwächerer Gewichtung von $NSim$ und $ESim$.

```
Solving Time (sec) : 42.54

objective value:                 0.890189540043963
v#9                                              1   (obj:0)
x#9#1#19                                         1   (obj:0)
x#9#3#4                                          1   (obj:0)
x#9#4#3                                          1   (obj:0)
x#9#7#17                                         1   (obj:0)
x#9#9#15                                         1   (obj:0)
x#9#10#12                                        1   (obj:0)
x#9#11#1                                         1   (obj:0)
x#9#14#14                                        1   (obj:0)
x#9#15#9                                         1   (obj:0)

ASim                                             1   (obj:0.2)
BSim                                             1   (obj:0.2)
penVSimPi                0.948148148148148       (obj:0.2)
penVSimOpt                                       1   (obj:0.2)
NSim                     0.473684210526316       (obj:0.1)
ESim                     0.531914893617021       (obj:0.1)
```

Dies kommt dem Goldstandard ziemlich nahe. Es ist hier nur ein Abbildungselement zu viel im Vergleich zum Goldstandard. Und zwar werden in dieser neunelementigen Abbildung noch die Aktivitäten mit der Beschreibung „Evaluate“ aufeinander abgebildet, die im Goldstandard nicht als korrespondierende Aktivitäten angegeben sind. Allerdings sind, was diese beiden Aktivitäten angeht, alle Eigenschaften (Beschriftung, Optionalität, Wiederholbarkeit usw.) bis auf die relative Position gleich, und die relative Position hat eine Ähnlichkeit von etwa $(1-|0,60-0,71|) = 0,89$, was ungefähr so hoch wie die optimale globale Ähnlichkeit bei der zuletzt angegebenen Gewichtung der Einzelähnlichkeiten ist. Deswegen wird diese Korrespondenz der „Evaluate“-Aktivitäten in kaum einer Abbildung nicht vorhanden sein. Es stellt sich hier vielmehr die Frage, warum diese Korrespondenz im Goldstandard nicht vorhanden ist, denn es handelt sich, wenn man die Modelle in Abbildung 5.8 und 5.9 betrachtet, bei beiden Aktivitäten um die Evaluation der Bewerbungsunterlagen, die bei der jeweiligen Universität eingehen. Insofern erkennt das Ganzzahlige Lineare Programm diese Korrespondenz richtig.

5.3.3 Güte der kalibrierten Zielfunktion

Die Beurteilung der Güte der gefundenen Abbildung kann über Genauigkeit (*precision*) und Trefferquote (*recall*) erfolgen. Die Genauigkeit gibt den Anteil der gefundenen positiven Treffer (*true positives*, tp) zur Anzahl aller Treffer (*true positives*+*false positives*, $tp + fp$) an und beantwortet damit die Frage: Wie viele der gefundenen Treffer sind relevante Treffer? Die Trefferquote gibt den Anteil der gefundenen positiven Treffer zur Anzahl aller relevanten Elemente (*true positives*+*false negatives*, $tp + fn$) an und beantwortet damit die Frage: Wie viele relevante Treffer wurden überhaupt gefunden?

- $precision(\cdot,\cdot) = \frac{tp}{tp+fp}$
- $recall(\cdot,\cdot) = \frac{tp}{tp+fn}$

True positives sind Treffer, die auch als solche erkannt wurden, *false positives* sind fälschlicher Weise als positiv erkannte Treffer, *false negatives* sind Treffer, die hätten erkannt werden müssen, dies aber nicht wurden. Unter der Gewichtung $w_{BSim} = 0,2$, $w_{VSim^\pi} = 0,2$, $w_{VSim^o} = 0,2$, $w_{ASim} = 0,2$, $w_{NSim} = 0,1$ und $w_{ESim} = 0,1$ werden die Prozessmodelle der neun verschiedenen Universitäten jeweils paarweise miteinander verglichen. Die Ergebnisse sind in Tabelle 5.4 abgebildet, wobei die Zahlen auf zwei Nachkommastellen gerundet sind. Das F-Maß stellt das gewichtete harmonische Mittel der Genauigkeit und der Trefferquote dar:

- $F = {}^{2\cdot(precision\cdot recall)}/_{(precision+recall)}$

Tabelle 5.4 zeigt die Genauigkeit und die Trefferquote für die paarweisen Vergleiche. Im Goldstandard werden zu den hier berücksichtigten Korrespondenzen weitere genannt (immer in Verbindung mit dem Modell der TU München), die jedoch jeweils eine Aktivität/ein Ereignis doppelt verwenden. Da diese Korrespondenzen per Definition mit der vorgestellten 1:1-Methode nicht gefunden werden können, wurden diese bei der Berechnung der Genauigkeit und der Trefferquote herausgenommen. Dies ist bei den mit * gekennzeichneten Zeilen der Fall.

Zum Vergleich liefern die im Rahmen des Process Model Matching Contest 2015 vorgestellten Abgleichsmechanismen Werte aus den Bereichen, wie sie in Tabelle 5.5 eingetragen sind. Der Macro-Durchschnitt ist jeweils das arithmetische Mittel von Genauigkeit, Trefferquote und F-Maß der einzelnen Abgleiche. Der Micro-Durchschnitt ist die Berechnung von Genauigkeit, Trefferquote und F-Maß über die Summe der *true positives*, *false positives* und *false negatives* aller Modellpaare zusammen. Der Micro-Durchschnitt berücksichtigt also unterschiedliche Größen der Testfälle. Bei der statischen Gewichtsverteilung fällt auf, dass die Trefferquote ziemlich hoch ist, also dass kaum ein *false negative* auftaucht, d. h., es werden fast alle Aktivitätenpaare erkannt, die erkannt werden sollen. Die Genauigkeit ist dagegen im Mittelmaß, da tendenziell zu viele Aktivitätenpaare als Treffer erkannt werden, also die Anzahl der *false positives* recht hoch ist. Dies könnte unter anderem daher rühren, dass nur zwei unterschiedliche Agenten an dem Prozess beteiligt sind, die in etwa gleich viele Aktivitäten auszuführen haben, die Wahrscheinlichkeit also bei etwa 50% liegt, dass die Agenten der Aktivitäten selbst bei zufälliger Zuordnung übereinstimmen. Das Ähnlichkeitsmaß $ASim$ hebt den Ähnlichkeitswert also sehr stark auch bei labeltechnisch nicht gut passenden Aktivitätenpaaren an. Da für $BSim$ jedoch ein einfaches Maß gewählt wurde, sollte $BSim$ auch nicht zu stark gewichtet werden. Insgesamt ist das Ergebnis als akzeptabel einzustufen, allerdings würde eine andere Wahl von $BSim$ und eine möglicherweise veränderte Gewichtung das Ergebnis bestimmt verbessern. Inwieweit eine Anpassung der Gewichte das Ergebnis beeinflussen kann, zeigt Abschnitt 5.3.4.

5.3.4 Dynamische Gewichtung der Perspektiven in der Zielfunktion

Bei der statischen Gewichtung werden in der Mehrheit der Fälle zu viele Knoten im Vergleich zum Goldstandard abgebildet. Da bei einem 1:1-Abgleich stark unterschiedlich granularer Prozessmodelle zwangsweise viele Knoten gelöscht werden, soll für diese Fälle der Anteil der abgebildeten Knoten und Kanten weniger stark Einfluss auf die Zielfunktion nehmen als für Modelle mit ähnlich großer Anzahl an Knoten. Das Gewicht von $NSim$ und $ESim$ wird so angepasst, dass es bei einer größeren Differenz der Knotenanzahl kleiner wird. Mit der Bezeichnung dk für die Differenz der Knotenanzahl wird das Gewicht von $NSim$ und $ESim$ jeweils auf $0,1 \cdot {}^1/\sqrt{dk+1}$ gesetzt. Da außerdem $penVSim^\pi$ und $penVSim^o$, die beide einen

Teil des Verhaltens beschreiben, im Vergleich zu $BSim$ und $ASim$ einzeln jeweils gleich stark in die Zielfunktion eingehen, wird auch ihre Gewichtung angepasst, sodass das Gewicht von $penVSim^{\pi}$ und $penVSim^{o}$ zusammen genommen so stark wie das von $BSim$ bzw. $ASim$ ist. Die Genauigkeit und die Trefferquote des paarweisen Abgleichs der neun Modelle ändern sich zu den Werten, die in Tabelle 5.6 dargestellt sind.

Mit dieser Art der Gewichtung ist die Anzahl der *false positives* deutlich verringert, nur noch etwa 16 % vom Wert mit statischer Gewichtung, allerdings hat sich die Anzahl der nicht erkannten Aktivitätenpaare etwa verdoppelt. Dies hat zur Folge, dass die Genauigkeit höher als bei der statischen Gewichtung ist, die Trefferquote jedoch geringer. Insgesamt ergibt dies für den Macro-Durchschnitt ein etwa gleichbleibendes F-Maß, während das F-Maß für den Micro-Durchschnitt einen deutlich besseren Wert liefert. Der Micro-Durchschnitt wird von den Initiatoren des Process Model Matching Contests als aussagekräftiger angesehen. Zur Berechnung des Macro-Durchschnitts des F-Maßes in Tabelle 5.6 wurden die mit NaN gekennzeichneten Felder als mit dem Wert 0 belegt einbezogen, was den üblichen Rechenregeln im Zusammenhang mit dem harmonischen Mittel genügt, wenn mindestens einer der Werte, über die gemittelt werden soll, 0 ist. Außerdem ist der harmonische Mittelwert immer kleiner gleich dem arithmetischen Mittelwert.

Aufgrund der unterschiedlichen Ergebnisse mit den verschiedenen Arten der Gewichtung fällt auf, dass der Prozessmodellabgleich nicht nur von den Ähnlichkeitsmaßen an sich abhängt, sondern auch stark von der Art der Gewichtung in der Zielfunktion aus Definition 4.33. Insgesamt liefert der vorgestellte Ansatz für die beiden Arten der Gewichtung keine, auf *precision* und *recall* bezogen, besseren Ergebnisse als bisherige Abgleichsansätze, jedoch wurde für die Implementierung zum einen die einfachste Methode zur Ähnlichkeitsbestimmung der Aktivitätenbeschreibung implementiert, nämlich die Levenshtein-Distanz, zum anderen liegt die Stärke des vorgestellten Abgleichsansatzes bei der Betrachtung von M:N-Korrespondenzen, die in der Evaluation mit den Modellen des Process Model Matching Contests aufgrund des zur Verfügung stehenden Goldstandards bzw. der Art der gegebenen Prozessmodelle nicht berücksichtigt wurden. Gerade im Falle der nicht erkannten Aktivitätenpaare ist bei einer elaborierteren Wahl von $BSim$, die beispielsweise Stemming beinhaltet und Synonyme erkennt, mit einer deutlichen Verbesserung der Ergebnisse zu rechnen. Dies wird in Abschnitt 6.2 im Rahmen des Ausblicks auf weiterführende Forschungsarbeiten noch einmal angesprochen.

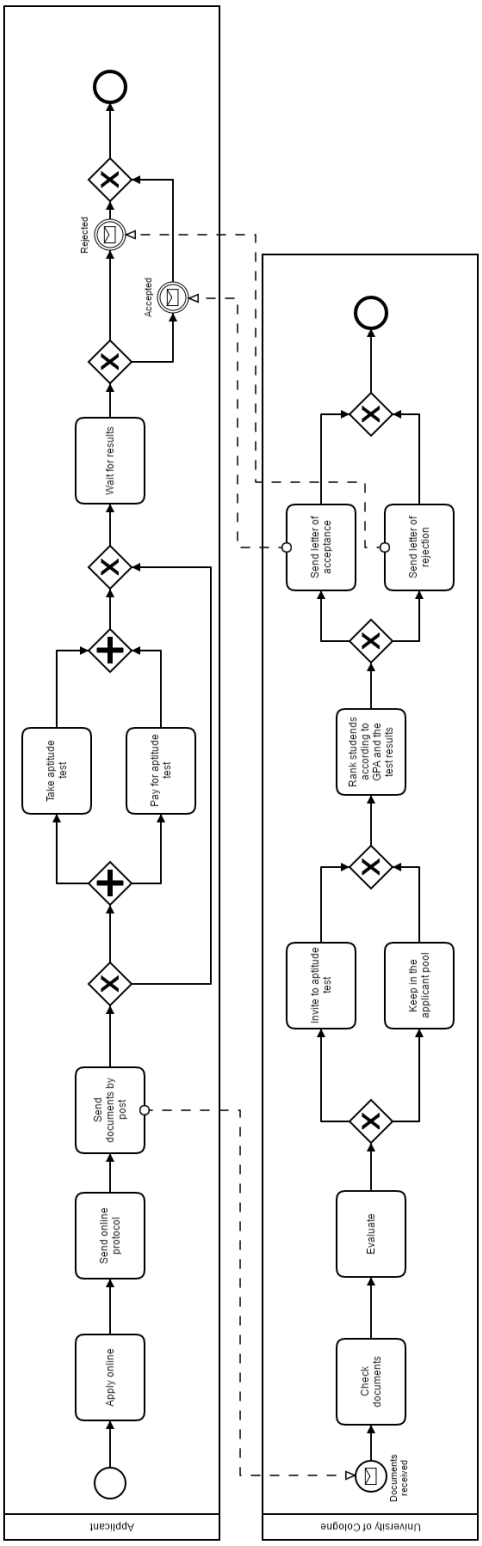


Abbildung 5.8: Beispielsmodell Bewerbungsprozess Uni Köln.

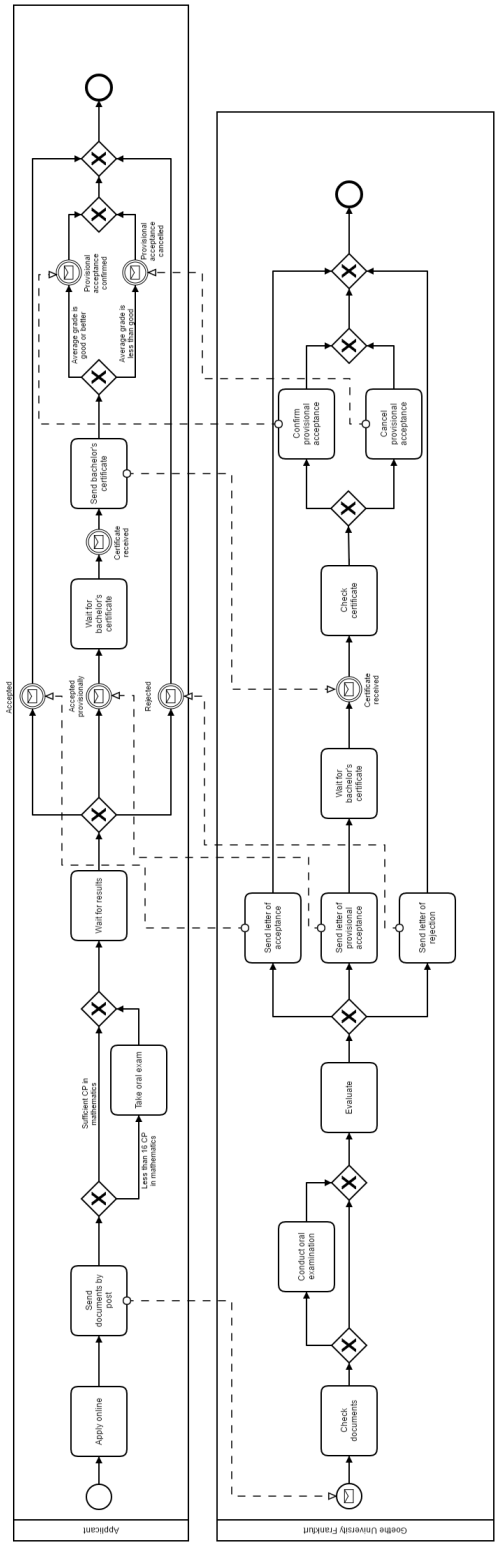


Abbildung 5.9: Beispielsmodell Bewerbungsprozess Uni Frankfurt.

Tabelle 5.4: Ergebnis des paarweisen Abgleichs der neun Prozessmodelle mit statischer Gewichtsverteilung.

Modellpaar	tp	fp	fn	precision	recall	F-Maß
Köln – Frankfurt	8	1	0	0,89	1,00	0,94
Köln – FU Berlin	4	10	0	0,29	1,00	0,44
Köln – Hohenheim	2	13	1	0,13	0,67	0,22
Köln – IIS Erlangen	2	13	0	0,13	1,00	0,24
Köln – Münster	2	9	0	0,18	1,00	0,31
Köln – Potsdam	1	14	1	0,07	0,5	0,12
Köln – TU München*	6	1	1	0,86	0,86	0,86
Köln – Würzburg	2	8	1	0,20	0,67	0,31
Frankfurt – FU Berlin	1	19	1	0,05	0,50	0,09
Frankfurt – Hohenheim	1	20	1	0,05	0,50	0,09
Frankfurt – IIS Erlangen	1	21	1	0,05	0,50	0,08
Frankfurt – Münster	1	22	0	0,04	1,00	0,08
Frankfurt – Potsdam	1	20	1	0,05	0,50	0,09
Frankfurt – TU München*	7	12	1	0,37	0,88	0,52
Frankfurt – Würzburg	2	16	1	0,11	0,67	0,19
FU Berlin – Hohenheim	4	18	1	0,18	0,80	0,30
FU Berlin – IIS Erlangen	14	8	0	0,64	1,00	0,78
FU Berlin – Münster	5	0	5	1,00	0,50	0,67
FU Berlin – Potsdam	13	9	1	0,59	0,93	0,72
FU Berlin –TU München*	2	16	2	0,11	0,50	0,18
FU Berlin – Würzburg	5	12	1	0,29	0,83	0,43
Hohenheim – IIS Erlangen	4	25	1	0,14	0,80	0,24
Hohenheim – Münster	5	21	3	0,19	0,63	0,29
Hohenheim – Potsdam	3	25	3	0,11	0,50	0,18
Hohenheim – TU München	1	18	4	0,05	0,20	0,08
Hohenheim – Würzburg	3	15	1	0,17	0,75	0,27
IIS Erlangen – Münster	8	19	5	0,30	0,62	0,40
IIS Erlangen – Potsdam	19	7	2	0,73	0,90	0,81
IIS Erlangen – TU München	2	17	5	0,11	0,29	0,15
IIS Erlangen – Würzburg	2	0	4	1,00	0,33	0,50
Münster – Potsdam	3	0	8	1,00	0,27	0,43
Münster – TU München	3	15	4	0,17	0,43	0,24
Münster – Würzburg	2	0	4	1,00	0,33	0,50
Potsdam – TU München*	1	18	3	0,05	0,25	0,09
Potsdam – Würzburg	5	13	1	0,28	0,83	0,42
TU München – Würzburg	2	16	2	0,11	0,50	0,18
Durchschnitt (Macro)				0,32	0,65	0,34
Durchschnitt (Micro)	79	471	70	0,15	0,53	0,23

Tabelle 5.5: Vergleichswerte (Minimum und Maximum) aus dem Matching Contest und eigene Werte mit statischer und dynamischer Gewichtung.

	Min	Max	statische Gewichtung	dynamische Gewichtung
precision (Macro)	0,125	0,855	0,32	0,60
recall (Macro)	0,292	0,626	0,65	0,30
F-Maß (Macro)	0,180	0,603	0,34	0,33
precision (Micro)	0,162	0,807	0,15	0,44
recall (Micro)	0,435	0,672	0,53	0,41
F-Maß (Micro)	0,253	0,668	0,23	0,42

Tabelle 5.6: Ergebnis des paarweisen Abgleichs der neun Prozessmodelle mit dynamischer Gewichtsverteilung.

Modellpaar	tp	fp	fn	precision	recall	F-Maß
Köln – Frankfurt	8	1	0	0,89	1,00	0,94
Köln – FU Berlin	1	0	3	1,00	0,25	0,40
Köln – Hohenheim	1	0	2	1,00	0,33	0,50
Köln – IIS Erlangen	0	2	2	0,00	0,00	NaN
Köln – Münster	0	1	2	0,00	0,00	NaN
Köln – Potsdam	0	2	2	0,00	0,00	NaN
Köln – TU München*	6	1	1	0,86	0,86	0,86
Köln – Würzburg	1	0	2	1,00	0,33	0,50
Frankfurt – FU Berlin	1	19	1	0,05	0,50	0,09
Frankfurt – Hohenheim	1	0	1	1,00	0,50	0,67
Frankfurt – IIS Erlangen	0	2	2	0,00	0,00	NaN
Frankfurt – Münster	0	1	1	0,00	0,00	NaN
Frankfurt – Potsdam	0	2	2	0,00	0,00	NaN
Frankfurt – TU München*	7	0	1	1,00	0,88	0,93
Frankfurt – Würzburg	1	0	2	1,00	0,33	0,50
FU Berlin – Hohenheim	1	1	4	0,50	0,20	0,29
FU Berlin – IIS Erlangen	13	0	1	1,00	0,93	0,96
FU Berlin – Münster	1	0	9	1,00	0,10	0,18
FU Berlin – Potsdam	13	0	1	1,00	0,93	0,96
FU Berlin –TU München*	2	16	2	0,11	0,50	0,18
FU Berlin – Würzburg	2	0	4	1,00	0,33	0,50
Hohenheim – IIS Erlangen	1	0	4	1,00	0,20	0,33
Hohenheim – Münster	1	0	7	1,00	0,125	0,22
Hohenheim – Potsdam	0	1	6	0,00	0,00	NaN
Hohenheim – TU München	0	1	5	0,00	0,00	NaN
Hohenheim – Würzburg	1	0	3	1,00	0,25	0,40
IIS Erlangen – Münster	1	0	12	1,00	0,08	0,14
IIS Erlangen – Potsdam	18	2	3	0,90	0,86	0,88
IIS Erlangen – TU München	1	2	6	0,33	0,14	0,20
IIS Erlangen – Würzburg	2	0	4	1,00	0,33	0,50
Münster – Potsdam	1	0	10	1,00	0,09	0,17
Münster – TU München	0	1	7	0,00	0,00	NaN
Münster – Würzburg	1	0	5	1,00	0,17	0,29
Potsdam – TU München*	0	2	4	0,00	0,00	NaN
Potsdam – Würzburg	1	0	5	1,00	0,17	0,29
TU München – Würzburg	2	17	2	0,11	0,50	0,17
Durchschnitt (Macro)				0,60	0,30	0,33
Durchschnitt (Micro)	89	74	128	0,44	0,41	0,42

Kapitel 6

Zusammenfassung und zukünftige Arbeiten

Zum Abschluss der Arbeit wird eine Zusammenfassung gegeben, die die Motivation, die Herausforderungen und die während der Arbeit entwickelten Methoden kurz noch einmal erläutert. Anschließend werden Einschränkungen, denen die entwickelten Methoden unterliegen, genannt und in diesem Zusammenhang ein Ausblick auf fortführende Forschungsthemen gegeben.

6.1 Zusammenfassung

Ein Ähnlichkeitsabgleich von Prozessmodellen ist vor allem zur Verwaltung großer Modellrepositorien, aber auch zur Konformitätsprüfung, zur Verbesserung der Verständlichkeit oder zum Einsatz bei Evaluationen hilfreich oder sogar notwendig. Dabei sind unter anderem eine meist nicht einheitliche Formulierung von Aufgaben und eine prinzipielle Freiheit in der Wahl der Feinheit der modellierten Schritte Probleme, die einen Abgleich schwierig machen. Dazu kommen generell unterschiedliche Modelliersprachen, die sich grob in imperative, d. h. flussorientierte, und deklarative, d. h. regelbasierte, Prozessmodelliersprachen und in diesen beiden Gruppen wiederum in verschiedene Sprachen unterscheiden lassen.

Um einen Ähnlichkeitsabgleich auf beliebigen Modellen durchzuführen und dabei die genannten Herausforderungen zu berücksichtigen, werden in der vorliegenden Arbeit geeignete Ähnlichkeitsmaße eingeführt, die diesen Zweck erfüllen. Zunächst werden bestehende Abgleichsmechanismen aus der verwandten Literatur betrachtet, die vornehmlich auf der Beschriftung von Aktivitäten, auf der Struktur von imperativen Prozessmodellen und auf dem Verhalten von Prozessmodellen aufbauen. Die Untersuchung dieser bestehenden Methoden legt auch offen, für welche Anwendungsfelder noch keine Methoden existieren, wobei die Anwendungsfelder in drei Dimensionen aufgespannt werden. Eine Dimension ist die Modellart der zu vergleichenden Modelle. Es können zwei imperative, zwei deklarative oder ein imperatives und ein deklaratives Modell verglichen werden. In den bisher erschienenen Arbeiten werden hauptsächlich imperative Prozessmodelle miteinander verglichen. Eine zweite Dimension ist die der Prozessperspektiven. Zusätzlich zur Beschriftung und zum Verhalten werden im Laufe der Arbeit auch zugewiesene Agenten, verwendete Datenobjekte und angesprochene Services beim Abgleich berücksichtigt. Die Struktur der Prozessmodelle, die teilweise im Verhalten widergespiegelt wird, wird mit Ausblick auf deklarative Prozessmodelle nicht weiter betrachtet. Die dritte Dimension betrifft die Abbildung, also die Korrespondenzenbildung

zwischen den Modellen. Um unterschiedliche Granularität der Prozessschritte beim Abgleich zu berücksichtigen, werden M:N-Abbildungen, d. h. Abbildungen, die Mengen von Aktivitäten auf Mengen von Aktivitäten abbilden, eingeführt. Gerade wenn die Verständlichkeit von Prozessmodelliersprachen untersucht wird, also wenn ein Prozessmodell in eines in einer anderen Sprache übersetzt wird und somit eine vorher bekannte 1:1-Beziehung zwischen den Aktivitäten besteht, sollten dennoch explizit 1:1-Abbildungen Verwendung finden, die tendenziell eine genauere Ähnlichkeitsbestimmung erlauben. In der verwandten Literatur finden sich vornehmlich 1:1-Abbildungen. Werden dort 1:N- und M:N-Abbildungen verwendet, so fehlt bislang eine klare Definition dieser und oftmals eine Anpassung bzw. Erweiterung der Abgleichsmethoden auf diese Arten der Abbildung.

Die Ähnlichkeitsbestimmung der funktionalen Perspektive, also der Aktivitätenbeschriftungen, ist in der Literatur ausreichend untersucht und eine Anpassung der Abgleichsmethoden auf M:N-Abbildungen ist mittels Konkatenation der Beschriftungen einfach zu bewerkstelligen. Die Methoden, die das Verhalten betreffen, lassen sich nicht ohne Weiteres auf M:N-Abbildungen erweitern. Hierfür werden in der vorliegenden Arbeit die Begriffe der mittleren, relativen Position, der mittleren Optionalität und der mittleren Wiederholbarkeit eingeführt, wobei mit relativer Position, Optionalität und Wiederholbarkeit das Verhalten als Merkmal einzelner Aktivitäten definiert wird. Ein Maß aufbauend auf (schwachen) Ordnungsrelationen ist ebenfalls angegeben, das jedoch Bedingungen an die zu vergleichenden Modelle stellt, um vernünftige Ergebnisse liefern zu können. Für explizite 1:1-Abbildungen werden Flussabhängigkeiten und ein darauf aufbauendes Ähnlichkeitsmaß betrachtet, das sowohl Reihenfolge als auch Kausalität zwischen einzelnen Aktivitäten berücksichtigt. Für die organisatorische Perspektive (Agenten), die datenorientierte Perspektive (Datenobjekte) und die operationale Perspektive (Services) existieren in der Literatur bislang keine Abgleichsmechanismen. Für diese drei Perspektiven werden jeweils ähnliche Ähnlichkeitsmaße, die auf einem Abgleich von Mengen basieren, vorgestellt. Anschließend wird die Übertragbarkeit der genannten Methoden auf deklarative Modelle untersucht, die über die Auffassung der Modelle in der generalisierten Form, die auf Mengen an Aktivitäten und Ressourcen und einer Menge an Relationen basiert, für fast alle vorgestellten Ähnlichkeitsmaße möglich ist.

Zum Abschluss wird ein Großteil der vorgestellten, neu entwickelten Ähnlichkeitsmaße im Rahmen eines Abgleichs der Modelle des Process Model Matching Contests (Antunes et al., 2015) verwendet, für die ein Goldstandard zur Überprüfung der Güte des Abgleichsansatzes vorliegt. Insgesamt liefert der Ansatz gute Ergebnisse, wobei sich der F-Wert als Ausgleichsmaß zwischen Trefferquote (recall) und Genauigkeit (precision) im oberen Mittelfeld der Teilnehmer des Contests aufhält. Da hier nur 1:1-Abgleiche vorgenommen werden, kann der M:N-Ansatz der vorliegenden Arbeit seine in Testsituationen überprüfte Stärke nicht voll entfalten. Für M:N-Abgleiche existiert aktuell jedoch kein Satz an Prozessmodellen mit einem vorgegebenen Goldstandard. Wie die Anwendung des Ansatzes auf die Modelle des Contests zeigt, sind die automatisch generierten Ergebnisse keine absoluten Aussagen sondern dienen in erster Linie als Hilfestellung für einen Modellier- und Domänenexperten für die weitere Bearbeitung, zum Beispiel der Zusammenführung der Modelle.

6.2 Einschränkungen und Fortführung der Forschung

Der vorgestellte Ansatz zum Abgleich von multiperspektivischen Prozessmodellen bietet in den folgenden, zum Teil schon im Zuge der jeweiligen Abschnitte genannten Bereichen Möglichkeiten zu einer Erweiterung bzw. Fortführung der Forschung.

- Wie in den Definitionen 3.1 und 3.2 des imperativen bzw. deklarativen, multiperspektivischen Prozessmodells ersichtlich, ist eine Ressourcenzuordnung in der organistorischen Perspektive so zu lesen, dass Agent A oder Agent B oder Agent C usw. eine Aufgabe ausführen kann ($A \vee B \vee C \vee \ldots$), während die Zuordnung in der datenorientierten und auch in der operationalen Perspektive als Konjunktion zu lesen ist: Objekt d_1 und Objekt d_2 und Objekt d_3 usw. wird für die Durchführung der Aktivität benötigt ($d_1 \wedge d_2 \wedge d_3 \wedge \ldots$). Entsprechend bieten sich auch für die Ähnlichkeitsmaße dieser Perspektiven unterschiedliche Varianten des Jaccard-Koeffizienten an. Um die Zuweisungen von Ressourcen flexibler zu gestalten, ist es vorstellbar, die Definition der multiperspektivischen Prozessmodelle dahingehend zu erweitern, als dass Ressourcenzuordnungen der Form $(R_1 \vee R_2 \vee \ldots \vee \ R_i) \wedge \ldots \wedge (R'_1 \vee R'_2 \vee \ldots \vee R'_j)$ zulässig sind. Dies würde beispielsweise die Zuweisung von Teams zu bestimmten Aktivitäten für eine kollaborative Ausführung (Cabanillas, 2016) erlauben (z. B. „Eine Untersuchung muss von einem Oberarzt (O) und entweder einer Krankenschwester (K) oder einem weiteren Arzt (A) durchgeführt werden“: $O \wedge (K \vee A)$). Grundsätzlich ist es auch denkbar, jede mögliche Ressourcenkonstellation pro Aktivität, mit der diese ausgeführt werden kann, anzugeben. Entsprechend müssen die Definitionen der jeweiligen Ähnlichkeitsmaße erweitert werden.

- Der Abgleichsansatz in dieser Arbeit fokussiert allein die Prozessmodelle und mögliche Ausführungen, jedoch keine tatsächlichen Prozessinstanzen. Wie Weidlich et al. (2008) schreiben, kann beispielsweise auch die Ausführungshäufigkeit bestimmter Aktivitäten bei einem Abgleich berücksichtigt werden um wichtige Pfade und Ausnahmen besser voneinander unterscheiden zu können.

- Die Implementierung kann, wie vor allem in Abschnitt 5.3 geschrieben, verbessert werden, indem beispielsweise zur Abspeicherung der Ähnlichkeitsmatrix keine CSV-Datei verwendet wird. Aufgrund der hyperexponentiell vielen Möglichkeiten an M:N-Abbildungen, die zwischen zwei Modellen gebildet werden können, stellt die Hardware die größten Probleme für die prototypische Implementierung dar. Auch für den M:N-Abgleich der Aktivitätsbeschriftungen wurde einer der einfachsten Ansätze aus der Literatur, die Levenshtein-Distanz, verwendet. Die Implementierung eines fortgeschritteneren Ansatzes würde die Güte der Ergebnisse mutmaßlich verbessern.

- Wie in der methodenzusammenfassenden Abbildung 4.19 ersichtlich, ist vor allem für den M:N-Abgleich des Verhaltens von deklarativen Prozessmodellen noch kein vollends zufriedenstellendes Ähnlichkeitsmaß gefunden. Zwar können Optionalität und Wiederholbarkeit für Mengen an Aktivitäten auch für deklarative Prozessmodelle bestimmt und abgeglichen werden, doch stellen diese beiden Merkmale nur einen Ausschnitt aus dem kompletten Modellverhalten dar. Die in Abschnitt 4.5.3 genannten Abgleichsmethoden (Ausführungspfade, Verhaltensmuster, Regelausdrücke, Flussabhängigkeiten) für deklarative Prozessmodelle unter 1:1-Abbildungen können näher untersucht und auf eine Erweiterung auf M:N-Abbildungen getestet werden.

- Wie in Abschnitt 5.3.4 geschrieben, hängt das Ergebnis des Abgleichs stark von der Gewichtsverteilung in der Zielfunktion aus Definition 4.33 ab. In Abschnitt 5.3 wurde die Abgleichsgüte für zwei mehr oder weniger beliebig gewählte Gewichtsverteilungen durchgeführt. Wenn eine ausreichend große Menge an Testmodellen mit optimalen Matches zur Verfügung steht, könnte jedoch eine genaue Kalibrierung der Gewichte durch-

geführt werden. Interessant wäre hierbei auch die Frage, ob die Gewichtsverteilung von bestimmten Merkmalen der Prozessmodelle abhängt, wobei diese Merkmale vor einem Abgleich natürlich bekannt sein müssen.

Kapitel A

Anhang

A.1 Distanzmaß und Metrik

Definition A.1 (Distanzmaß)**.** Eine Funktion $d : X \times X \to [0, \infty)$ ist ein Distanzmaß auf einer Menge X, falls sie den folgenden Eigenschaften für $x, y \in X$ genügt:

- Nicht-Negativität: $d(x, y) \geq 0$
- Symmetrie: $d(x, y) = d(y, x)$
- Identität: $d(x, y) = 0 \Leftrightarrow x = y$

Definition A.2 (Metrik)**.** Eine Funktion $d : X \times X \to [0, \infty)$ ist eine Metrik, wenn d ein Distanzmaß ist und zusätzlich die Dreiecksungleichung erfüllt:

- Dreiecksungleichung: $d(x, z) \leq d(x, y) + d(y, z)\ \forall x, y, z \in X$

Von einer Pseudometrik spricht man dann, wenn d alle Eigenschaften einer Metrik erfüllt, mit Ausnahme der Identität, die bei einer Pseudometrik nur in eine Richtung gelten muss: $d(x, x) = 0$ (d. h., $x = y \Rightarrow d(x, y) = 0$).

A.2 Algorithmus zur Bestimmung der Optionalität von Knoten

In den Abbildungen A.1 bis A.7 ist ein beispielhafter Durchlauf des Algorithmus aus Abschnitt 4.3.3 zur Bestimmung der Optionalität von Knoten dargestellt. Die Erklärungen der einzelnen Schritte finden sich in den jeweiligen Bildunterschriften.

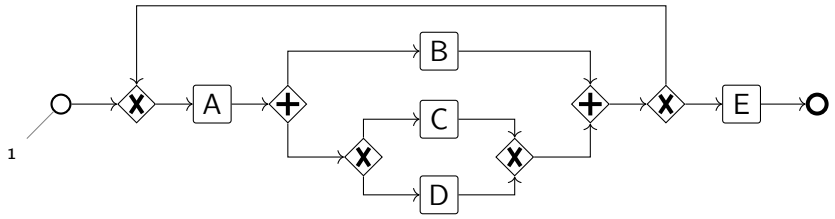


Abbildung A.1: Startmarkierung (tag=1) am Startknoten.

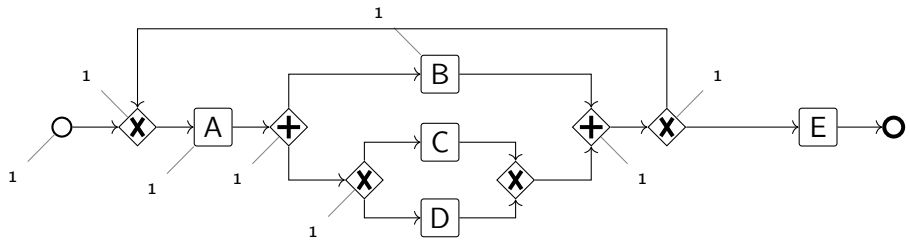


Abbildung A.2: Startmarkierung (tag=1) wird so lange weitergeschickt, bis sie auf XOR-Splits trifft

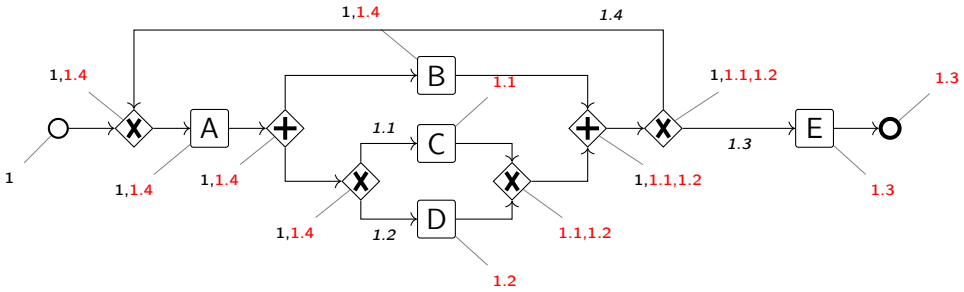


Abbildung A.3: Eindeutige Erweiterungen der Markierungen nach XOR-Splits (1 → 1.1, 1.2 und 1 → 1.3, 1.4), Weiterschicken dieser, bis sie auf das nächste XOR-Split treffen und Überprüfen, ob Obermengenbedingung an den XOR-Splits für die neu hinzugefügten Markierungen zutreffen. Hier: 1.4 ⊉ 1.1, 1,2 und 1.1 ⊉ 1.3, 1.4 und 1.2 ⊉ 1.3, 1.4.

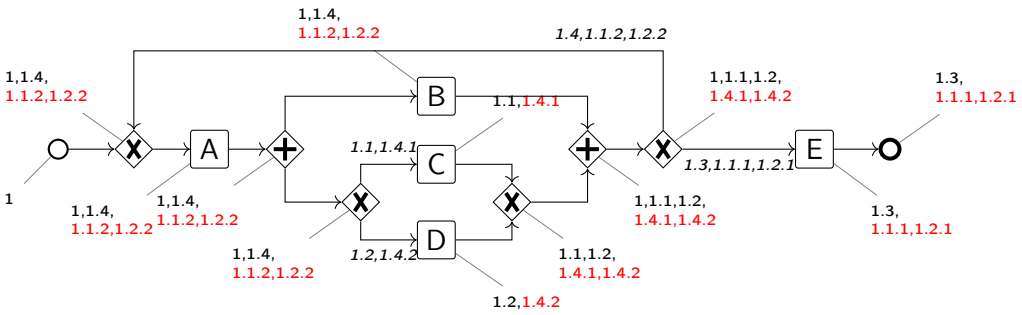


Abbildung A.4: Eindeutige Erweiterungen der Markierungen nach XOR-Splits (1.4 → 1.4.1, 1.4.2 und 1.1 → 1.1.1, 1.1.2 und 1.2 → 1.2.1, 1.2.2), Weiterschicken dieser, bis sie auf das nächste XOR-Split treffen und Überprüfen, ob Obermengenbedingung an den XOR-Splits für die neu hinzugefügten Markierungen zutreffen. Hier: 1.1.2 ⊇ 1.1 und 1.2.2 ⊇ 1.2 sowie 1.4.1 ⊇ 1.4 und 1.4.2 ⊇ 1.4. D. h., alle neuen Markierungen werden von den XOR-Splits unverändert weitergegeben.

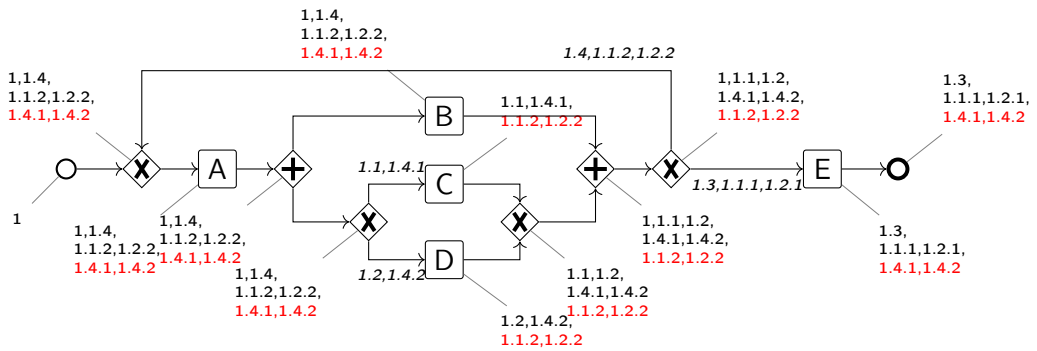


Abbildung A.5: Überprüfung bei XOR-Splits ergibt, dass 1.4.1 ⊇ 1.4.1 und 1.4.2 ⊇ 1.4.2 und 1.1.2 ⊇ 1.1.2 und 1.2.2 ⊇ 1.2.2. D. h., neu hinzugefügte Markierungen werden unverändert weitergeschickt.

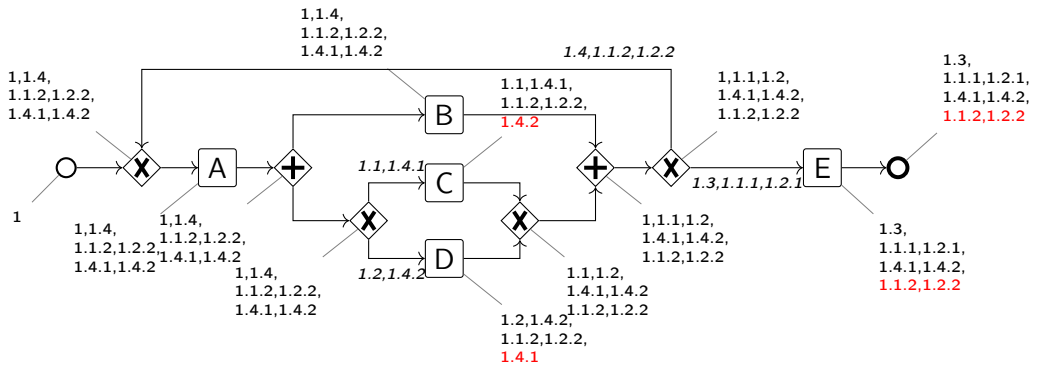


Abbildung A.6: Weiterschicken so lange, bis neue Markierungen auf Knoten treffen, die bereits mit derselben Markierung versehen sind, oder bis zum Endereignis. Die Belegung der Knoten mit Markierungen ist nach diesem Durchlauf beendet.

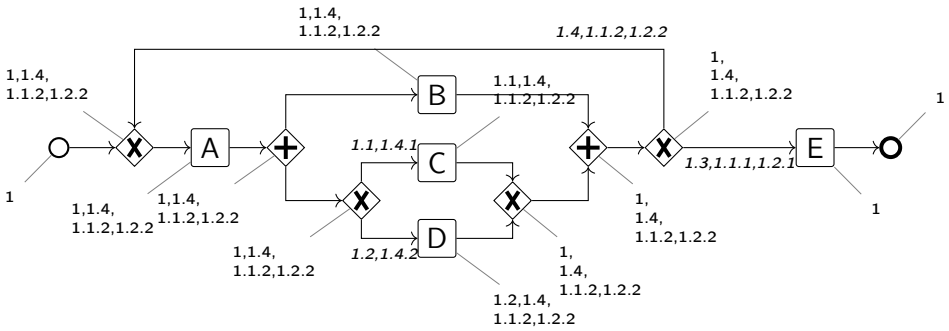


Abbildung A.7: Nach Zusammenfassen der Markierungen: Aktivitäten C und D sind die einzigen Knoten ohne Startmarkierung, d. h., sie sind optional.

A.3 Anzahl aller möglichen M:N-Abbildungen

Für eine n-elementige Menge gibt es

$$S_{n,k} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n$$

Möglichkeiten, diese in k nichtleere, disjunkte Teilmengen zu zerlegen. Diese Zahl wird Stirling Zahl zweiter Art (Aigner, 2007) genannt. Sie gibt also an, wie viele Möglichkeiten es gibt, ein Prozessmodell in k Aktivitätenmengen zu zerlegen. Der Parameter k kann hierbei Werte zwischen 1 und der Anzahl an Aktivitäten im Prozessmodell erreichen. Ist $k = 1$, so werden alle Aktivitäten zu einer Aktivitätenmenge zusammengefasst, ist $k = |N_i|$ für Prozessmodell G_i, so wird jede Aktivität in eine separate, einelementige Aktivitätenmenge gepackt. Insgesamt kann k aber bis zum Wert $k = |N_i| + 1$ gehen, wenn die leere Menge Teil der Partition ist. Daraus folgt, dass es

$$\eta_j = \sum_{k=1}^{|N_i|} 2 \cdot S_{|N_i|,k}$$

mögliche Partitionen von N_i gibt, wobei das Malnehmen mit 2 daher rührt, dass einmal für alle Möglichkeiten die leere Menge nicht mit dabei ist (sie wird bei der Stirling-Zahl nicht mitgezählt) und einmal schon.

Um nun die Anzahl der Abbildungen zu bestimmen, wird folgende Feststellung gemacht: Die Mächtigkeit der Partitionen beider Prozessmodelle muss immer gleich sein. Es sei k die Mächtigkeit der Partitionen. Wird keine Aktivität gelöscht, so bestehen beide Partitionen aus k nichtleeren Aktivitätenmengen (die leere Menge ist in keiner Partition enthalten). Wird nur aus einem Modell etwas gelöscht, so besteht eine Partition aus k nicht-leeren Aktivitätenmengen (wovon eine Aktivitätenmenge die gelöschte ist) und die andere Partition aus $k - 1$ nichtleeren Aktivitätenmenge und der leeren Menge. Wird aus beiden Modellen mindestens jeweils eine Aktivität gelöscht, so bestehen beide Partitionen jeweils aus $k - 1$ nichtleeren Aktivitätenmengen und der leeren Menge.

- Wird aus keinem Modell etwas gelöscht, so gibt es, wenn die Abbildung k Elemente enthält, $S_{|N_1|,k} \cdot S_{|N_2|,k}$ viele Möglichkeiten für Partitionen in beiden Modellen und $S_{|N_1|,k} \cdot S_{|N_2|,k} \cdot k!$ viele Zuordnungsmöglichkeiten.

- Wird aus dem ersten Modell etwas gelöscht, aber aus dem zweiten nicht, so hat das erste Modell k-viele (nichtleere) Partitionselemente und das zweite $k-1$-viele nichtleere Partitionselemente. Die Partition von Modell 2 hat, mit der leeren Menge, Mächtigkeit k. Das heißt, es gibt insgesamt $S_{|N_1|,k} \cdot S_{|N_2|,k-1}$ viele mögliche Partitionen in beiden Modellen. Abbildungsmöglichkeiten bei einer festen Partition gibt es, wie im ersten Fall auch, $k!$ viele (hier kann die leere Menge als normales Partitionselement angesehen werden), was zu $S_{|N_1|,k} \cdot S_{|N_2|,k-1} \cdot k!$ vielen Abbildungsmöglichkeiten führt.

- Im Fall, dass nur aus dem zweiten Modell etwas gelöscht wird, erfolgt die Überlegung analog zum vorherigen Fall, d. h., es gibt $S_{|N_1|,k-1} \cdot S_{|N_2|,k} \cdot k!$ viele Zuordnungsmöglichkeiten für den Fall, dass nur aus dem ersten Modell Aktivitäten nicht abgebildet werden.

- Für den letzten Fall, den, dass aus beiden Modellen etwas gelöscht wird, gibt es $S_{|N_1|,k-1} \cdot S_{|N_2|,k-1}$ viele verschiedene Partitionen. Bei der Anzahl der Abbildungsmöglichkeiten muss nun der Fall ausgeschlossen werden, dass die leere Menge auf die leere Menge abgebildet wird, denn es soll aus beiden Modellen mindestens eine Aktivität gelöscht werden, d. h., es muss immer ein nichtleeres Partitionselement der leeren Menge zugewiesen werden. Aus den grundsätzlich $k!$ vielen Zuordnungsmöglichkeiten für jeweils eine feste Partition müssen die, die die leere Menge auf die leere Menge abbilden, herausgenommen werden. Wird festgehalten, dass die leere Menge auf die leere Menge abgebildet wird, so gibt es für die jeweils $k-1$ vielen nichtleeren Mengen $(k-1)!$ viele Zuordnungsmöglichkeiten. Es wird also $(k-1)!$ von $k!$ abgezogen, um die Anzahl aller möglichen Zuordnung ohne diejenigen, die die leere Menge auf die leere Menge abbilden, zu bekommen. Dies führt zu $S_{|N_1|,k-1} \cdot S_{|N_2|,k-1} \cdot (k! - (k-1)!)$ vielen Abbildungsmöglichkeiten.

Die gerade genannten Fälle werden nun summiert und über alle möglichen k zusammengezählt. Der Parameter k (die Mächtigkeit der Partitionen) kann dabei von 1 bis zur Anzahl der kleineren Aktivitätenmenge plus 1 (alle Aktivitäten einzeln plus die leere Menge) gehen; $N_{min} = \min\{|N_1|, |N_2|\}$. Es ergibt sich für die Anzahl aller möglicher M:N-Abbildungen zwischen zwei Prozessmodellen

$$\eta = \sum_{k=1}^{N_{min}+1} \Big(\big(S_{|N_1|,k} \cdot S_{|N_2|,k} + S_{|N_1|,k-1} \cdot S_{|N_2|,k} + S_{|N_1|,k} \cdot S_{|N_2|,k-1} \big) \cdot k! \\ + S_{|N_1|,k-1} \cdot S_{|N_2|,k-1} \cdot (k! - (k-1)!) \Big),$$

was einer „kombinatorischen Detonation“ gleichkommt.

A.4 Beispielprozessmodelle 2, 3 und 4 für Expertenbefragung

In diesem Abschnitt finden sich die Beispielprozessmodelle R2 (Abbildung A.8), R3 (Abbildung A.9) und R4 (Abbildung A.10) der Expertenbefragung aus Abschnitt 5.1.2 und die jeweils drei Varianten, von denen die ähnlichste und die unähnlichste in der Umfrage anzugeben war.

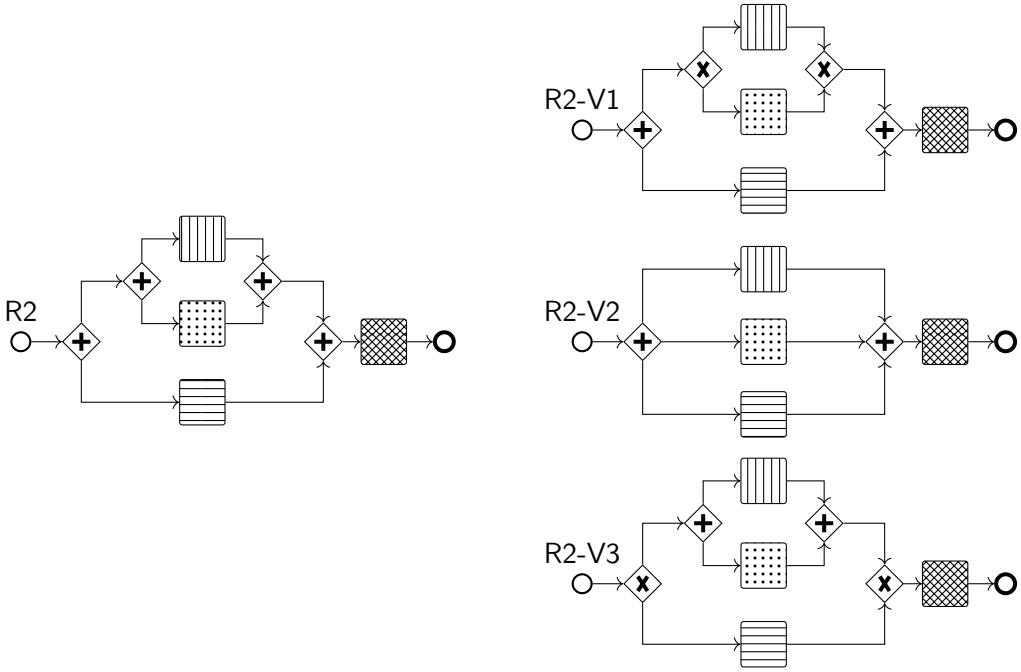


Abbildung A.8: Referenzmodell 2 (links) und drei Varianten (von oben nach unten Variante 1 bis 3).

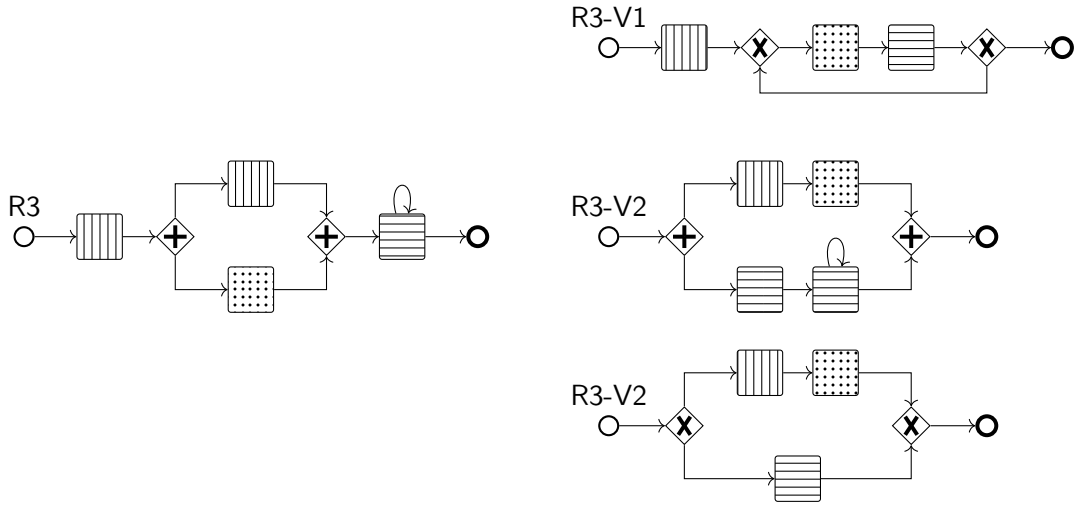


Abbildung A.9: Referenzmodell 3 (links) und drei Varianten (von oben nach unten Variante 1 bis 3).

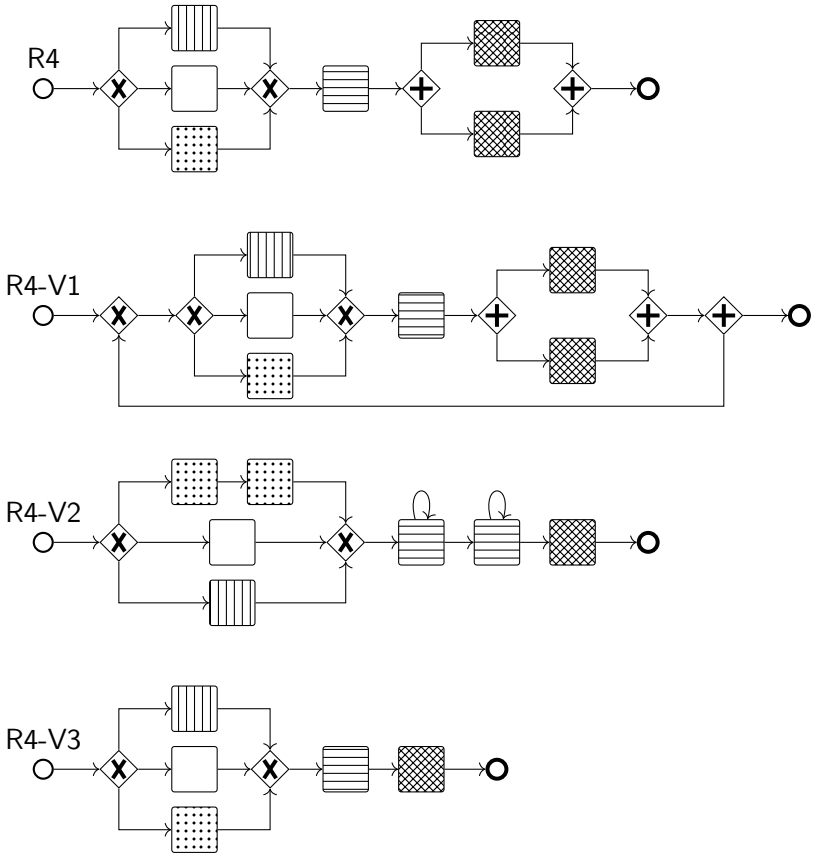


Abbildung A.10: Referenzmodell 4 (oben) und drei Varianten darunter (von oben nach unten Variante 1 bis 3).

A.5 ZIMPL-Programmcode

Der Code zum ZIMPL-Programm zum Finden der besten Abbildung, also der Abbildung, die die Zielfunktion, den Ähnlichkeitswert, maximiert, ist in Listing A.1 gezeigt.

Listing A.1: ZIMPL-Programmcode für bijektive M:N-Abbildungen.

```
###################################
# Similarity Process Modelling
# M:N-Abbildung
# Susanne Hoffmeister, Michaela Baumann
###################################

########## Model Graphs ##################
# Knoten einlesen
set n0 := { read "**PfadKnotendatei1**" as "<1s>" };
set n1 := { read "**PfadKnotendatei2**" as "<1s>" };

# Potenzmengen und zugehoerige Indexmengen generieren
set P0[] := powerset(n0);
set J0 := indexset(P0);
set P1[] := powerset(n1);
set J1 := indexset(P1);

# Bogen einlesen
param arcsG0[n0*n0] := read "**PfadKantendatei1**" as "<1s,2s> 3n" default 0;
param arcsG1[n1*n1] := read "**PfadKantendatei2**" as "<1s,2s> 3n" default 0;

defnumb hasArcG0(i,j) := if card(P0[i])>0 and card(P0[j])>0 and
  sum <nod1,nod2> in P0[i]*P0[j]: arcsG0[nod1,nod2] >= 1 then 1 else 0 end;
defnumb hasArcG1(i,j) := if card(P1[i])>0 and card(P1[j])>0 and
  sum <nod1,nod2> in P1[i]*P1[j]: arcsG1[nod1,nod2] >= 1 then 1 else 0 end;

# Einlesen der Matrix nach Index der Teilmenge in der Potenzmenge
# WICHTIG: Teilmengen in der Matrix muessen nach gleichem kanonischen Schema
# erzeugt sein
param BSimParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 3n" default 0;
param ASimParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 4n" default 0;
param VSimPiParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 5n" default 0;
param VSimPiPenParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 6n" default 0;
param VSimOParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 7n" default 0;
param VSimOPenParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 8n" default 0;
param VSimRhoParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 9n" default 0;
param VSimRhoPenParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 10n" default 0;
param DSimParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 11n" default 0;

# Obergrenze fuer Anzahl der abgebildeten TM
param M := min(card(n0),card(n1))+1;
set K :={1..M};
# Auswahlvariable (wie viele Abbildungselemente)
var v[K] binary;
# Zuordnung Menge j0 wird j0 zugeordnet bei einer Abb. mit k TM
var x[K*J0*J1] binary;

var aG0[J0*J0] binary;    # es ex. eine Kante von TM j zu j' in G0
var aG1[J1*J1] binary;    # es ex. eine Kante von TM j zu j' in G1

# Obergrenze Gesamtanzahl der Bogen
param M2 := 16;
set K2 := {1..M2};
# Auswahlvariable (wie viele zaehlende Kanten)
var Bogenanzahl[K2] binary;
# Binaervariable ob zaehlende Kanten ueberhaupt vorhanden
var eta binary;
# Kante von j nach j' in G0 ist gute Kante in Abb
var goodArcsG0[K2*J0*J0] binary;
# Kante von j nach j' in G1 ist gute Kante in Abb
```

```
var goodArcsG1[K2*J1*J1] binary;

var ASim;
var BSim;
var DSim;
var penVSimPi;
var penVSimOpt;
var penVSimRho;
var NSim; # ist "einfach" da Gesamtanzahl an Knoten durch Graph gegeben
var ESim;

maximize similarity: 1/8*(ASim + BSim + penVSimPi + penVSimOpt
      + penVSimRho + DSim + NSim + ESim);

# A Sim
subto penASim_equals:
  ASim ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
    x[k,j0,j1] * ASimParam[j0,j1]/k;

# B Sim
subto penBSim_equals:
  BSim ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
    x[k,j0,j1] * BSimParam[j0,j1]/k;

# Node Position
subto penVSimPi_equals:
  penVSimPi ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
    x[k,j0,j1] * max((VSimPiParam[j0,j1]-VSimPiPenParam[j0,j1]),0)/k;

# Node Optionality
subto penVSimOpt_equals:
  penVSimOpt ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
    x[k,j0,j1] * max((VSimOParam[j0,j1]-VSimOPenParam[j0,j1]),0)/k;

# Node Repeatability
subto penVSimRho_equals:
  penVSimRho ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
    x[k,j0,j1] * max((VSimRhoParam[j0,j1]-VSimRhoPenParam[j0,j1]),0)/k;

# D Sim
subto penDSim_equals:
  DSim ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
    x[k,j0,j1] * DSimParam[j0,j1]/k;

subto NSim_equals:
  NSim ==
    1 - (sum <k,j0> in K*J0: x[k,j0,0]*card(P0[j0])
    + sum <k,j1> in K*J1: x[k,0,j1]*card(P1[j1]))/(card(n0)+card(n1));

subto ESim_equals:
  ESim ==
    sum <k,i1,i2> in K2*J0*J0: 1/k*goodArcsG0[k,i1,i2]
    + sum <k,j1,j2> in K2*J1*J1: 1/k*goodArcsG1[k,j1,j2] + eta;

#########################
# Nebenbedingung fuer Teilmengen
#########################
# Kopplung Knoten in Teilmenge
subto is_node_in_subset_of_P0:
  forall <n> in n0:
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0] inter {n})!=0:
```

```
      x[k,j0,j1]==1;

subto is_node_in_subset_of_P1:
  forall <n> in n1:
    sum <k,j0,j1> in K*J0*J1 with card(P1[j1] inter {n})!=0:
      x[k,j0,j1]==1;

# leere Menge darf auch nur maximal einmal vorkommen
subto is_empty_subset_of_P0:
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0]) == 0:
      x[k,j0,j1] <= 1;

subto is_empty_subset_of_P1:
    sum <k,j0,j1> in K*J0*J1 with card(P1[j1]) == 0:
      x[k,j0,j1] <= 1;

# leere Menge darf nicht auf leere Menge abgebildet werden
subto not_empty_set_on_empty_set:
  forall <k> in K:
    x[k,0,0] == 0;


#########################
# Kopplung v-x, k-Sets werden zugeordnet
#########################

subto mapping_has_at_least_k_nodes:
  forall <k> in K:
    k*v[k] == sum <j0,j1> in J0*J1: x[k,j0,j1];

subto eine_vk_auf_eins:
  sum <k> in K:
    v[k] == 1;

#########################
# ESim
#########################

subto kante_zwischen_TM_vorhanden_in_G0:
  forall <i1,i2> in J0*J0:
    aG0[i1,i2] <= hasArcG0(i1,i2);

subto kante_zwischen_TM_vorhanden_in_G1:
  forall <i1,i2> in J1*J1:
    aG1[i1,i2] <= hasArcG1(i1,i2);

subto kante_zaehlt_in_G0a:
  forall <i1,i2> in J0*J0:
    1/2*(sum <k,j> in K*J1 : x[k,i1,j]
      + sum <k,j> in K*J1 : x[k,i2,j]) >= aG0[i1,i2];

subto kante_zaehlt_in_G0b:
  forall <i1,i2> in J0*J0:
    (sum <k,j> in K*J1 : x[k,i1,j]
      + sum <k,j> in K*J1 : x[k,i2,j]-1)*hasArcG0(i1,i2) <= aG0[i1,i2];

subto kante_zaehlt_in_G1a:
  forall <i1,i2> in J1*J1:
    1/2*(sum <k,j> in K*J0 : x[k,j,i1]
      + sum <k,j> in K*J0 : x[k,j,i2]) >= aG1[i1,i2];

subto kante_zaehlt_in_G1b:
  forall <i1,i2> in J1*J1:
    (sum <k,j> in K*J0 : x[k,j,i1]
      + sum <k,j> in K*J0 : x[k,j,i2]-1)*hasArcG1(i1,i2) <= aG1[i1,i2];

subto kantenanzahl_bestimmen:
```

```
  sum <k> in K2:
  k*Bogenanzahl[k] == sum <i1,i2> in J0*J0: aG0[i1,i2]
    + sum <j1,j2> in J1*J1: aG1[j1,j2];

subto nur_maximal_eine_Bogenanzahl:
  sum <k> in K2:
    Bogenanzahl[k] <= 1;

subto Bogenanzahl_null:
  1 - sum <k> in K2: Bogenanzahl[k] >= eta;

subto guteBogen_G0:
  forall <k,i1,i2> in K2*J0*J0:
    goodArcsG0[k,i1,i2] <= aG0[i1,i2];

subto guteBogen_G0b:
  forall <i1,i2> in J0*J0:
    sum<k> in K2:
    2*goodArcsG0[k,i1,i2] <=
      sum <j1,j2> in J1*J1:
      ((sum<k> in K: x[k,i1,j1] + sum<k> in K: x[k,i2,j2])*hasArcG1(j1,j2));

subto guteBogen_G1:
  forall <k,i1,i2> in K2*J1*J1:
    goodArcsG1[k,i1,i2] <= aG1[i1,i2];

subto guteBogen_G1b:
  forall <i1,i2> in J1*J1:
    sum<k> in K2:
      2*goodArcsG1[k,i1,i2] <= sum <j1,j2> in J0*J0:
      ((sum<k> in K: x[k,j1,i1] + sum<k> in K: x[k,j2,i2])*hasArcG0(j1,j2));

# Kopplung goodArcs-Bogenanzahl insgesamt werden zugeordnet
subto goodArcs_k2Bogen_G0:
  forall <k,i,j> in K2*J0*J0:
    Bogenanzahl[k] >= goodArcsG0[k,i,j];

subto goodArcs_k2Bogen_G1:
  forall <k,i,j> in K2*J1*J1:
    Bogenanzahl[k] >= goodArcsG1[k,i,j];
```

Speziell für 1:1-Abbildungen muss der ZIMPL-Code angepasst werden. Eine modifizierte, deutlich kürzere Version ist in Listing A.2 gegeben.

Listing A.2: ZIMPL-Programmcode für partiell injektive 1:1-Abbildungen.

```
####################################
# Similarity Process Modelling
# 1:1-Abbildung
# Susanne Hoffmeister, Michaela Baumann
####################################

########## Model Graphs ##################
# Knoten einlesen
# set J0 := {0..15};
set n0 := {read "**PfadKnotendatei1**" as "<1s>"};
set J0 := {0..card(n0)};
set P0[J0]:= <0> {},
# hier: Liste der Zuordnung Index-Aktivitaet von G1 der Form
# <Nr> {"Aktivitaet_ID"},
# beginnend bei 1 einfuegen und mit ; statt letztem Komma abschliessen

set n1 := {read "**PfadKnotendatei2**" as "<1s>"};
set J1 := {0..card(n1)};
set P1[J1] := <0> {},
# hier: Liste der Zuordnung Index-Aktivitaet von G2 der Form
# <Nr> {"Aktivitaet_ID"},
# beginnend bei 1 einfuegen und mit ; statt letztem Komma abschliessen

# keine automatische Generierung der Potenzmenge (wird nicht benoetigt)
# deswegen Einlesen der Liste in vorgegebener Form

# Bogen einlesen
param arcsG0[n0*n0] := read "**PfadKantendatei1**" as "<1s,2s> 3n" default 0;
param arcsG1[n1*n1] := read "**PfadKantendatei2**" as "<1s,2s> 3n" default 0;

defnumb hasArcG0(i,j) := if card(P0[i])>0 and card(P0[j])>0 and
  sum <nod1,nod2> in P0[i]*P0[j]: arcsG0[nod1,nod2] >= 1 then 1 else 0 end;
defnumb hasArcG1(i,j) := if card(P1[i])>0 and card(P1[j])>0 and
  sum <nod1,nod2> in P1[i]*P1[j]: arcsG1[nod1,nod2] >= 1 then 1 else 0 end;

param kantenG0 := sum<j0,j1>in J0*J0: hasArcG0(j0,j1);
param kantenG1 := sum<j0,j1>in J1*J1: hasArcG1(j0,j1);
param gesamtKanten := kantenG0 + kantenG1;

# Einlesen der Matrix nach Index der Teilmenge in der Potenzmenge
# WICHTIG: Teilmengen in der Matrix muessen nach gleichem kanonischen Schema
# erzeugt sein
param BSimParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 3n" default 0;
param ASimParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 4n" default 0;
param VSimPiParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 5n" default 0;
param VSimPiPenParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 6n" default 0;
param VSimOParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 7n" default 0;
param VSimOPenParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 8n" default 0;
param VSimRhoParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 9n" default 0;
param VSimRhoPenParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 10n" default 0;
param DSimParam[J0*J1] := read "**PfadSimdatei**" as "<1n,2n> 11n" default 0;

# Obergrenze fuer Anzahl der abgebildeten TM
param M := min(card(n0),card(n1))+1;
set K :={1..M};
# Auswahlvariable
var v[K] binary;
# Zuordnung Menge j0 wird j0 zugeordnet bei einer Abb. mit k TM
var x[K*J0*J1] binary;

var goodArcsG0[J0*J0] binary;   # Kante von j nach j' in G0 ist gute Kante in Abb
var goodArcsG1[J1*J1] binary; # Kante von j nach j' in G1 ist gute Kante in Abb
```

```
var ASim;
var BSim;
var DSim;
var penVSimPi;
var penVSimOpt;
var penVSimRho;
var NSim;          # Gesamtanzahl an Knoten durch Graph gegeben
var ESim;          # Gesamtzahl an Kanten durch Graph gegeben (bei 1:1-Abbildung)

maximize similarity: 1/8*(ASim + BSim + penVSimPi + penVSimOpt
  + penVSimRho + DSim + NSim + ESim);

# A Sim
subto penASim_equals:
  ASim ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
      x[k,j0,j1] * ASimParam[j0,j1]/k;

# B Sim
subto penBSim_equals:
  BSim ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
      x[k,j0,j1] * BSimParam[j0,j1]/k;

# Node Position
subto penVSimPi_equals:
  penVSimPi ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
      x[k,j0,j1] * max((VSimPiParam[j0,j1] - VSimPiPenParam[j0,j1]),0)/k;

# Node Optionality
subto penVSimOpt_equals:
  penVSimOpt ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
      x[k,j0,j1] * max((VSimOParam[j0,j1] - VSimOPenParam[j0,j1]),0)/k;

# Node Repeatability
subto penVSimRho_equals:
  penVSimRho ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
      x[k,j0,j1] * max((VSimRhoParam[j0,j1] - VSimRhoPenParam[j0,j1]),0)/k;

# D Sim
subto penDSim_equals:
  DSim ==
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0])>0 and card(P1[j1])>0 :
      x[k,j0,j1] * DSimParam[j0,j1]/k;

# NSim als Anzahl x[k,i,j] ohne diejenigen, die auf die leere Menge
# abgebildet werden
subto NSim_equals:
  NSim == ((sum <k,j0,j1> in K*J0*J1: 2*x[k,j0,j1]-sum<k,j1>in K*J1:
    x[k,0,j1]-sum<k,j0>in K*J0:x[k,j0,0])/(card(n0)+card(n1));

# ESim direkt als Anzahl abgebildeter Kanten (gute Kanten)
# durch Anzahl aller Kanten
subto ESim_equals:
  ESim == sum <i1,i2> in J0*J0: 1/gesamtKanten*goodArcsG0[i1,i2]
          + sum <j1,j2> in J1*J1: 1/gesamtKanten*goodArcsG1[j1,j2];

#########################
# Nebenbedingung fuer Teilmengen
#########################

# Kopplung Knoten in Teilmenge
subto is_node_in_subset_of_P0:
```

```
    forall <n> in n0:
      sum <k,j0,j1> in K*J0*J1 with card(P0[j0] inter {n}) != 0: x[k,j0,j1] <= 1;

subto is_node_in_subset_of_P1:
  forall <n> in n1:
    sum <k,j0,j1> in K*J0*J1 with card(P1[j1] inter {n}) != 0: x[k,j0,j1] <= 1;

# leere Menge darf auch nur maximal einmal vorkommen
subto is_empty_subset_of_P0:
    sum <k,j0,j1> in K*J0*J1 with card(P0[j0]) == 0: x[k,j0,j1] <= 1;

subto is_empty_subset_of_P1:
    sum <k,j0,j1> in K*J0*J1 with card(P1[j1]) == 0: x[k,j0,j1] <= 1;

# leere Menge darf nicht auf leere Menge abgebildet werden
subto not_empty_set_on_empty_set:
  forall <k> in K:
    x[k,0,0] == 0;

##########################
# Kopplung v-x, k-Sets werden zugeordnet
##########################

subto mapping_has_at_least_k_nodes:
  forall <k> in K:
    k*v[k] == sum <j0,j1> in J0*J1: x[k,j0,j1];

subto eine_vk_auf_eins:
  sum <k> in K:
    v[k] == 1;

##########################
# ESim
##########################

subto guteBogen_G0:
  forall <i1,i2> in J0*J0:
    goodArcsG0[i1,i2] <= hasArcG0(i1,i2);

subto guteBogen_G0b:
  forall <i1,i2> in J0*J0:
    2*goodArcsG0[i1,i2] <= sum <j1,j2> in J1*J1:
    ((sum<k> in K: x[k,i1,j1] + sum<k> in K: x[k,i2,j2])*hasArcG1(j1,j2));

subto guteBogen_G1:
  forall <i1,i2> in J1*J1:
    goodArcsG1[i1,i2] <= hasArcG1(i1,i2);

subto guteBogen_G1b:
  forall <i1,i2> in J1*J1:
    2*goodArcsG1[i1,i2] <= sum <j1,j2> in J0*J0:
    ((sum<k> in K: x[k,j1,i1] + sum<k> in K: x[k,j2,i2])*hasArcG0(j1,j2));
```

Literaturverzeichnis

L. Ackermann. *Sprachzentrierte Ansätze zur Steigerung der Akzeptanz von Geschäftsprozessmodellen*. Doktorarbeit, Universität Bayreuth (2017).

L. Ackermann, S. Schönig und S. Jablonski. *Simulation of Multi-perspective Declarative Process Models*, S. 61–73. Springer International Publishing, Cham (2017a). ISBN 978-3-319-58457-7. doi:10.1007/978-3-319-58457-7_5. URL http://dx.doi.org/10.1007/978-3-319-58457-7_5.

L. Ackermann, S. Schönig und S. Jablonski. *Towards Simulation- and Mining-based Translation of Resource-aware Process Models*, S. 359–371. Springer International Publishing, Cham (2017b). ISBN 978-3-319-58457-7. doi:10.1007/978-3-319-58457-7_26. URL http://dx.doi.org/10.1007/978-3-319-58457-7_26.

M. Aigner. *Diskrete Mathematik*. Springer-Verlag (2007).

F. Aiolli, A. Burattin und A. Sperduti. *A Business Process Metric Based on the Alpha Algorithm Relations*, S. 141–146. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). ISBN 978-3-642-28108-2. doi:10.1007/978-3-642-28108-2_13. URL http://dx.doi.org/10.1007/978-3-642-28108-2_13.

R. Akkiraju und A. Ivan. *Discovering Business Process Similarities: An Empirical Study with SAP Best Practice Business Processes*, S. 515–526. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). ISBN 978-3-642-17358-5. doi:10.1007/978-3-642-17358-5_35. URL http://dx.doi.org/10.1007/978-3-642-17358-5_35.

A. Alves de Medeiros, W. van der Aalst und A. Weijters. Quantifying process equivalence based on observed behavior. *Data & Knowledge Engineering*, 64(1), S. 55 – 74 (2008). ISSN 0169-023X. doi:http://doi.org/10.1016/j.datak.2007.06.010. URL http://www.sciencedirect.com/science/article/pii/S0169023X07001206. Fourth International Conference on Business Process Management (BPM 2006)8th International Conference on Enterprise Information Systems (ICEIS' 2006)Four selected and extended papers-Three selected and extended papers.

G. Antoniou und F. van Harmelen. *Web Ontology Language: OWL*, S. 67–92. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). ISBN 978-3-540-24750-0. doi:10.1007/978-3-540-24750-0_4. URL http://dx.doi.org/10.1007/978-3-540-24750-0_4.

G. Antunes et al. The process model matching contest 2015. In *Enterprise Modelling and Information Systems Architectures*, Lecture Notes in Informatics (LNI), S. 127–155. Gesellschaft für Informatik, Bonn (2015).

A. Awad, A. Polyvyanyy und M. Weske. Semantic Querying of Business Process Models. In *2008 12th International IEEE Enterprise Distributed Object Computing Conference*, S. 85–94 (2008). ISSN 1541-7719. doi:10.1109/EDOC.2008.11.

J. Bae, J. Caverlee, L. Liu und H. Yan. *Process Mining by Measuring Process Block Similarity*, S. 141–152. Springer Berlin Heidelberg, Berlin, Heidelberg (2006a). ISBN 978-3-540-38445-8. doi:10.1007/11837862_15. URL `http://dx.doi.org/10.1007/11837862_15`.

J. Bae, L. Liu, J. Caverlee und W. B. Rouse. Process Mining, Discovery, and Integration using Distance Measures. In *2006 IEEE International Conference on Web Services (ICWS'06)*, S. 479–488 (2006b). doi:10.1109/ICWS.2006.105.

M. Baumann. Comparing Imperative and Declarative Process Models (2017). URL `https://epub.uni-bayreuth.de/3325/`. Preprint.

M. Baumann, M. H. Baumann, L. Ackermann, S. Schönig und S. Jablonski. Ansätze zum Ähnlichkeitsabgleich von deklarativen Geschäftsprozessmodellen. In *INFORMATIK 2016*, Band 259 von *Proceedings / GI-Edition*, S. 733–738. Köllen, Bonn (2016a). URL `http://subs.emis.de/LNI/Proceedings/Proceedings259/733.pdf`.

M. Baumann, M. H. Baumann, D. F.-X. Gruber und S. Jablonski. Infinite Horizon Decision Support For Rule-based Process Models. *International Journal on Advances in Software*, 9(1-2), S. 141–153 (2016b). URL `http://www.iariajournals.org/software/soft%5fv9%5fn12%5f2016%5fpaged.pdf`.

M. Baumann, M. H. Baumann und S. Jablonski. On Behavioral Process Model Similarity Matching: A Centroid-based Approach. In *ICCGI 2015, The Tenth International Multi-Conference on Computing in the Global Information Technology*, Band 5, S. 125–131 (2015a).

M. Baumann, M. H. Baumann und S. Jablonski. On Behavioral Process Model Similarity Matching: A Centroid-based Approach (Enlarged Abstract of [BBJ15]). In *INFORMATIK 2016*, Band 259 von *Proceedings / GI-Edition*, S. 731–732. Köllen, Bonn (2016c). URL `http://subs.emis.de/LNI/Proceedings/Proceedings259/731.pdf`.

M. Baumann, M. H. Baumann, S. Schönig und S. Jablonski. Resource-Aware Process Model Similarity Matching. In F. Toumani und et al. (Hg.), *Service-Oriented Computing - ICSOC 2014 Workshops*, Band 8954 von *Lecture Notes in Computer Science*, S. 96–107. Springer International Publishing (2015b). ISBN 978-3-319-22884-6. doi:10.1007/978-3-319-22885-3_9.

M. H. Baumann, M. Baumann, S. Schönig und S. Jablonski. Towards Multi-perspective Process Model Similarity Matching. In J. Barjis und R. Pergl (Hg.), *Enterprise and Organizational Modeling and Simulation*, Band 191 von *LNBIP*, S. 21–37. Springer Berlin Heidelberg (2014). ISBN 978-3-662-44859-5.

M. Becker und R. Laue. A comparative survey of business process similarity measures. *Computers in Industry*, 63(2), S. 148–67 (2012). ISSN 0166-3615.

E. Bertino, E. Ferrari und V. Atluri. The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *ACM Trans. Inf. Syst. Secur.*, 2(1), S.

65–104 (1999). ISSN 1094-9224. doi:10.1145/300830.300837. URL `http://doi.acm.org/10.1145/300830.300837`.

G. Bisson. Learning in FOL with a similarity measure. In *Proc. of the Nat. Conf. on Artificial Intelligence*, S. 82–82 (1992).

E. L. Bradley. *Overlapping Coefficient*. John Wiley & Sons, Inc. (2004). ISBN 9780471667193. doi:10.1002/0471667196.ess1900. URL `http://dx.doi.org/10.1002/0471667196.ess1900`.

M. Braschler und B. Ripplinger. How Effective is Stemming and Decompounding for German Text Retrieval? *Information Retrieval*, 7(3), S. 291–316 (2004). ISSN 1573-7659. doi:10.1023/B:INRT.0000011208.60754.a1. URL `https://doi.org/10.1023/B:INRT.0000011208.60754.a1`.

J. R. Büchi. *On a Decision Method in Restricted Second Order Arithmetic*, S. 425–435. Springer New York, New York, NY (1990). ISBN 978-1-4613-8928-6. doi:10.1007/978-1-4613-8928-6_23. URL `http://dx.doi.org/10.1007/978-1-4613-8928-6_23`.

H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8), S. 689 – 694 (1997). ISSN 0167-8655. doi:http://dx.doi.org/10.1016/S0167-8655(97)00060-3. URL `http://www.sciencedirect.com/science/article/pii/S0167865597000603`.

H. Bunke. Recent developments in graph matching. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Band 2, S. 117–124 vol.2 (2000). ISSN 1051-4651. doi:10.1109/ICPR.2000.906030.

H. Bunke und X. Jiang. *Graph Matching and Similarity*, S. 281–304. Springer US, Boston, MA (2000). ISBN 978-1-4615-4401-2. doi:10.1007/978-1-4615-4401-2_10. URL `http://dx.doi.org/10.1007/978-1-4615-4401-2_10`.

C. Bussler. *Organisationsverwaltung in Workflow-Management-Systemen*. DUV Springer Fachmedien Wiesbaden GmbH (1998).

C. Cabanillas. Process- and Resource-Aware Information Systems. In *2016 IEEE 20th International Enterprise Distributed Object Computing Conference (EDOC)*, S. 1–10 (2016). doi:10.1109/EDOC.2016.7579383.

M. Castelo Branco, J. Troya, K. Czarnecki, J. Küster und H. Völzer. *Matching Business Process Workflows across Abstraction Levels*, S. 626–641. Springer Berlin Heidelberg, Berlin, Heidelberg (2012a). ISBN 978-3-642-33666-9. doi:10.1007/978-3-642-33666-9_40. URL `http://dx.doi.org/10.1007/978-3-642-33666-9_40`.

M. Castelo Branco, Y. Xiong, K. Czarnecki, J. Küster und H. Völzer. An Empirical Study on Consistency Management of Business and IT Process Models. *GSDLAB TECHNICAL REPORT, Waterloo* (2012b).

U. Cayoglu et al. The process model matching contest 2013. In *4th International Workshop on Process Model Collections: Management and Reuse, PMC-MR* (2013).

M. Cheatham und P. Hitzler. *String Similarity Metrics for Ontology Alignment*, S. 294–309. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). ISBN 978-3-642-41338-4. doi:10.1007/978-3-642-41338-4_19. URL https://doi.org/10.1007/978-3-642-41338-4_19.

W. Cohen, P. Ravikumar und S. Fienberg. A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, Band 3, S. 73–78 (2003).

B. Curtis, M. I. Kellner und J. Over. Process Modeling. *Commun. ACM*, 35(9), S. 75–90 (1992). ISSN 0001-0782. doi:10.1145/130994.130998. URL http://doi.acm.org/10.1145/130994.130998.

J. De Smedt, J. De Weerdt, E. Serral und J. Vathienen. *Improving Understandability of Declarative Process Models by Revealing Hidden Dependencies*, S. 83–98. Springer International Publishing, Cham (2016). ISBN 978-3-319-39696-5. doi:10.1007/978-3-319-39696-5_6. URL https://Doi.org/10.1007/978-3-319-39696-5_6.

R. Dijkman. A Classification of Differences between Similar BusinessProcesses. In *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*, S. 37–37 (2007). ISSN 1541-7719. doi:10.1109/EDOC.2007.24.

R. Dijkman, M. Dumas und L. García-Bañuelos. Graph Matching Algorithms for Business Process Model Similarity Search. In U. Dayal, J. Eder, J. Koehler und H. A. Reijers (Hg.), *Business Process Management*, Band 5701 von *LNCS*, S. 48–63. Springer Berlin Heidelberg (2009a). ISBN 978-3-642-03847-1.

R. Dijkman, M. Dumas, L. García-Bañuelos und R. Käärik. Aligning Business Process Models. In *International Enterprise Distributed Object Computing Conference*, S. 45–53. IEEE (2009b). ISBN 978-0-7695-3785-6.

R. Dijkman, M. Dumas, B. van Dongen, R. Käärik und J. Mendling. Similarity of business process models: Metrics and evaluation. *Information Systems*, 36(2), S. 498 – 516 (2011). ISSN 0306-4379.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), S. 269–271 (1959). ISSN 0945-3245. doi:10.1007/BF01386390. URL http://dx.doi.org/10.1007/BF01386390.

P. Dourish, J. Holmes, A. MacLean, P. Marqvardsen und A. Zbyslaw. Freeflow: Mediating Between Representation and Action in Workflow Systems. In *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work*, CSCW '96, S. 190–198. ACM (1996). ISBN 0-89791-765-0.

M. Dumas, M. La Rosa, J. Mendling und H. A. Reijers. *Fundamentals of Business Process Management*. Springer (2013). ISBN 978-3-642-33142-8.

M. Ehrig, A. Koschmider und A. Oberweis. Measuring Similarity Between Semantic Business Process Models. In *Proceedings of the Fourth Asia-Pacific Conference on Comceptual Modelling - Volume 67*, APCCM '07, S. 71–80. Australian Computer Society, Inc. (2007). ISBN 1-920-68285-X. URL http://dl.acm.org/citation.cfm?id=1274453.1274465.

C. C. Ekanayake, M. Dumas, L. García-Bañuelos, M. La Rosa und A. H. M. ter Hofstede. Approximate Clone Detection in Repositories of Business Process Models. In A. Barros, A. Gal und E. Kindler (Hg.), *BPM 2012: Proceedings*, S. 302–318. Springer Berlin Heidelberg (2012). ISBN 978-3-642-32885-5.

D. Fahland, D. Lübke, J. Mendling, H. Reijers, B. Weber, M. Weidlich und S. Zugal. Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer und R. Ukor (Hg.), *Enterprise, Business-Process and Information Systems Modeling*, Band 29 von *LNBIP*, S. 353–366. Springer Berlin Heidelberg (2009a). ISBN 978-3-642-01861-9. doi: 10.1007/978-3-642-01862-6_29.

D. Fahland, D. Lübke, J. Mendling, H. Reijers, B. Weber, M. Weidlich und S. Zugal. Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt, P. Soffer und R. Ukor (Hg.), *Enterprise, Business-Process and Information Systems Modeling*, Band 29 von *LNBIP*, S. 353–366. Springer Berlin Heidelberg (2009b). ISBN 978-3-642-01861-9. doi:10.1007/978-3-642-01862-6_29. URL `http://dx.doi.org/10.1007/978-3-642-01862-6_29`.

D. Fahland, J. Mendling, H. A. Reijers, B. Weber, M. Weidlich und S. Zugal. Declarative versus Imperative Process Modeling Languages: The Issue of Maintainability. In S. Rinderle-Ma, S. Sadiq und F. Leymann (Hg.), *Business Process Management Workshops*, Band 43 von *LNBIP*, S. 477–488. Springer Berlin Heidelberg (2010). ISBN 978-3-642-12185-2. doi: 10.1007/978-3-642-12186-9_45.

B. Fluri, M. Wuersch, M. PInzger und H. Gall. Change Distilling:Tree Differencing for Fine-Grained Source Code Change Extraction. *IEEE Transactions on Software Engineering*, 33(11), S. 725–743 (2007). ISSN 0098-5589. doi:10.1109/TSE.2007.70731.

S. Fortin. The Graph Isomorphism Problem. Technischer Bericht, University of Alberta (1996). doi:doi:10.7939/R3SX64C5K.

U. Frank. Towards a pluralistic conception of research methods in information systems research. ICB-Research Report 7, Essen (2006). URL `http://hdl.handle.net/10419/58156`.

B. J. Frey und D. Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315(5814), S. 972–976 (2007). ISSN 0036-8075. doi:10.1126/science.1136800. URL `http://science.sciencemag.org/content/315/5814/972`.

M. Gaitanides. *Prozeßorganisation*. Verlag Franz Vahlen GmbH (2012).

D. Giannakopoulou und K. Havelund. Automata-based verification of temporal properties on running programs. In *16th Annual Int. Conf. on Automated Software Engineering (ASE)*, S. 412–416 (2001). ISSN 1938-4300.

B. Goethals. *Apriori Property and Breadth-First Search Algorithms*, S. 124–127. Springer US, Boston, MA (2009). ISBN 978-0-387-39940-9. doi:10.1007/978-0-387-39940-9_23. URL `http://dx.doi.org/10.1007/978-0-387-39940-9_23`.

D. Grigori, J. C. Corrales, M. Bouzeghoub und A. Gater. Ranking BPEL Processes for Service Discovery. *IEEE Transactions on Services Computing*, 3(3), S. 178–192 (2010). ISSN 1939-1374. doi:10.1109/TSC.2010.6.

C. W. Günther und W. M. P. van der Aalst. *Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics*, S. 328–343. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). ISBN 978-3-540-75183-0. doi:10.1007/978-3-540-75183-0_24. URL http://dx.doi.org/10.1007/978-3-540-75183-0_24.

S. Hallé, R. Villemaire, O. Cherkaoui und R. Deca. *A Logical Approach to Data-Aware Automated Sequence Generation*, S. 192–216. Springer Berlin Heidelberg (2012). ISBN 978-3-642-28525-7.

R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 29(2), S. 147–160 (1950). ISSN 1538-7305. doi:10.1002/j.1538-7305.1950.tb00463.x. URL http://dx.doi.org/10.1002/j.1538-7305.1950.tb00463.x.

E. Hellinger. Neue Begründung der Theorie quadratischer Formen von unendlich vielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 136, S. 210–271 (1909).

W. Hesse. Ontologie(n). *Informatik-Spektrum*, 25(6), S. 477–480 (2002). ISSN 1432-122X. doi:10.1007/s002870200265. URL http://dx.doi.org/10.1007/s002870200265.

A. Hevner, S. March, J. Park und S. Ram. *Design Science Research in Information Systems*, S. 9–22. Springer US (2010). ISBN 978-1-4419-5653-8.

J. Hidders, M. Dumas, W. M. P. van der Aalst, A. H. M. ter Hofstede und J. Verelst. When Are Two Workflows the Same? In *Proceedings of the 2005 Australasian Symposium on Theory of Computing - Volume 41*, CATS '05, S. 3–11. Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2005). ISBN 1-920682-23-6. URL http://dl.acm.org/citation.cfm?id=1082260.1082261.

R. A. Horn und C. R. Johnson. *Matrix analysis*. Cambridge university press (2012).

A. Huang. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, S. 49–56 (2008).

K. Huang, Z. Zhou, Y. Han, G. Li und J. Wang. *An Algorithm for Calculating Process Similarity to Cluster Open-Source Process Designs*, S. 107–114. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). ISBN 978-3-540-30207-0. doi:10.1007/978-3-540-30207-0_14. URL http://dx.doi.org/10.1007/978-3-540-30207-0_14.

S. Jablonski. MOBILE: A modular workflow model and architecture. In *Proc. of Int. Working Conference on Dynamic Modelling and Information Systems, Nordwijkerhout*. citeseer (1994).

S. Jablonski. Do We Really Know How to Support Processes? Considerations and Reconstruction. In G. Engels, C. Lewerentz, W. Schäfer, A. Schürr und B. Westfechtel (Hg.), *Graph Transformations and Model-Driven Engineering*, Band 5765 von *LNCS*, S. 393–410. Springer Berlin Heidelberg (2010). ISBN 978-3-642-17321-9. doi:10.1007/978-3-642-17322-6_17.

S. Jablonski und C. Bussler. *Workflow management: modeling concepts, architecture and implementation*. International Thomson Computer Press (1996). ISBN 978-1-850-32222-1.

M. A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406), S. 414–420 (1989). doi:10.1080/01621459.1989.10478785. URL `http://www.tandfonline.com/doi/abs/10.1080/01621459.1989.10478785`.

K. Jensen. *Coloured Petri nets: basic concepts, analysis methods and practical use*, Band 1. Springer Science & Business Media (2013).

R. Johnson, D. Pearson und K. Pingali. The Program Structure Tree: Computing Control Regions in Linear Time. In *Proceedings of the ACM SIGPLAN 1994 Conference on Programming Language Design and Implementation*, PLDI '94, S. 171–185. ACM, New York, NY, USA (1994). ISBN 0-89791-662-X. doi:10.1145/178243.178258. URL `http://doi.acm.org/10.1145/178243.178258`.

J. A. W. Kamp. Tense logic and the theory of linear order (1968).

E. Kasanen, K. Lukka und A. Siitonen. The Constructive Approach in Management Accounting Research. *Journal of Management Accounting Research*, 5, S. 243 – 264 (1993). ISSN 10492127. URL `http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=9701211561&site=ehost-live`.

M. G. Kendall. A New Measure of Rank Correlation. *Biometrika*, 30(1/2), S. 81–93 (1938). ISSN 00063444. URL `http://www.jstor.org/stable/2332226`.

M. G. Kendall. Rank correlation methods. *Charles Griffin & Company Limited* (1948).

C. Klinkmüller, H. Leopold, I. Weber, J. Mendling und A. Ludwig. Listen to Me: Improving Process Model Matching through User Feedback. In S. Sadiq, P. Soffer und H. Völzer (Hg.), *Business Process Management: Proceedings*, S. 84–100. Springer International Publishing (2014). ISBN 978-3-319-10172-9.

C. Klinkmüller, I. Weber, J. Mendling, H. Leopold und A. Ludwig. Increasing Recall of Process Model Matching by Improved Activity Label Matching. In F. Daniel, J. Wang und B. Weber (Hg.), *Business Process Management*, Band 8094 von *LNCS*, S. 211–218. Springer Berlin Heidelberg (2013). ISBN 978-3-642-40175-6.

A. Koschmider und A. Oberweis. Ontology Based Business Process Description. In *EMOI-INTEROP*, S. 321–333 (2005).

M. Kunze, M. Weidlich und M. Weske. Behavioral Similarity – A Proper Metric. In S. Rinderle-Ma, F. Toumani und K. Wolf (Hg.), *BPM 2011. Proceedings*, S. 166–181. Springer Berlin Heidelberg (2011). ISBN 978-3-642-23059-2.

M. Kunze und M. Weske. Metric Trees for Efficient Similarity Search in Large Process Model Repositories. In M. zur Muehlen und J. Su (Hg.), *Business Process Management Workshops*, Band 66 von *LNBIP*, S. 535–546. Springer Berlin Heidelberg (2011). ISBN 978-3-642-20510-1.

V. Künzle und M. Reichert. *Towards Object-Aware Process Management Systems: Issues, Challenges, Benefits*, S. 197–210. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). ISBN 978-3-642-01862-6. doi:10.1007/978-3-642-01862-6_17. URL `http://dx.doi.org/10.1007/978-3-642-01862-6_17`.

M. La Rosa, M. Dumas, A. H. ter Hofstede und J. Mendling. Configurable multi-perspective business process models. *Information Systems*, 36(2), S. 313 – 340 (2011). ISSN 0306-4379. doi:http://dx.doi.org/10.1016/j.is.2010.07.001. URL `http://www.sciencedirect.com/science/article/pii/S0306437910000633`. Special Issue: Semantic Integration of Data, Multimedia, and Services.

M. La Rosa, M. Dumas, R. Uba und R. Dijkman. *Merging Business Process Models*, S. 96–113. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). ISBN 978-3-642-16934-2. doi:10.1007/978-3-642-16934-2_10. URL `http://dx.doi.org/10.1007/978-3-642-16934-2_10`.

H. Leopold, J. Mendling und A. Polyvyanyy. Supporting Process Model Validation through Natural Language Generation. *IEEE Transactions on Software Engineering*, 40(8), S. 818–840 (2014). ISSN 0098-5589. doi:10.1109/TSE.2014.2327044.

C. Li, M. Reichert und A. Wombacher. *On Measuring Process Model Similarity Based on High-Level Change Operations*, S. 248–264. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). ISBN 978-3-540-87877-3. doi:10.1007/978-3-540-87877-3_19. URL `http://dx.doi.org/10.1007/978-3-540-87877-3_19`.

R. Liu, K. Bhattacharya und F. Y. Wu. *Modeling Business Contexture and Behavior Using Business Artifacts*, S. 324–339. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). ISBN 978-3-540-72988-4. doi:10.1007/978-3-540-72988-4_23. URL `http://dx.doi.org/10.1007/978-3-540-72988-4_23`.

J. B. Lovins. *Development of a stemming algorithm*. MIT Information Processing Group, Electronic Systems Laboratory Cambridge (1968).

J. Lu und F. Gao. Process Modeling Based on Process Similarity. *Industrial & Engineering Chemistry Research*, 47(6), S. 1967–1974 (2008). doi:10.1021/ie0704851. URL `http://dx.doi.org/10.1021/ie0704851`.

A. Maedche und S. Staab. *Measuring Similarity between Ontologies*, S. 251–263. Springer Berlin Heidelberg, Berlin, Heidelberg (2002). ISBN 978-3-540-45810-4. doi:10.1007/3-540-45810-7_24. URL `http://dx.doi.org/10.1007/3-540-45810-7_24`.

G. A. Miller. WordNet: A Lexical Database for English (1995).

M. Minor, A. Tartakovski und R. Bergmann. *Representation and Structure-Based Similarity Assessment for Agile Workflows*, S. 224–238. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). ISBN 978-3-540-74141-1. doi:10.1007/978-3-540-74141-1_16. URL `http://dx.doi.org/10.1007/978-3-540-74141-1_16`.

M. Montali, F. Chesani, P. Mello und F. M. Maggi. Towards Data-aware Constraints in Declare. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, S. 1391–1396. ACM, New York, NY, USA (2013). ISBN 978-1-4503-1656-9. doi:10.1145/2480362.2480624. URL `http://doi.acm.org/10.1145/2480362.2480624`.

R. R. Mukkamala. *A formal model for declarative workflows: dynamic condition response graphs*. Doktorarbeit, University of Copenhagen (2012).

OASIS. Business Process Execution Language 2.0 (2007). URL `http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html[accessed:2017-06-02]`.

Object Management Group. Business Process Model and Notation 2.0 (2011). URL `http://www.omg.org/spec/BPMN/2.0/PDF/[accessed:2017-02-18]`.

Object Management Group. Case Management Model and Notation Version 1.0 (2014). URL `http://www.omg.org/spec/CMMN/1.0/PDF/[accessed:2015-07-19]`.

A. Oyegoke. The constructive research approach in project management research. *International Journal of Managing Projects in Business*, 4(4), S. 573–595 (2011). doi:10.1108/17538371111164029. URL `http://dx.doi.org/10.1108/17538371111164029`.

C. D. Paice. An Evaluation Method for Stemming Algorithms. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, S. 42–50. Springer-Verlag New York, Inc. (1994). ISBN 0-387-19889-X. URL `http://dl.acm.org/citation.cfm?id=188490.188499`.

M. Pesic. *Contraint-Based Workflow Management Systems: Shifting Control to Users*. Doktorarbeit, Technische Universiteit Eindhoven (2008).

M. Pesic, H. Schonenberg und W. van der Aalst. DECLARE: Full Support for Loosely-Structured Processes. In *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*, S. 287–287 (2007). ISSN 1541-7719.

M. Pesic und W. van der Aalst. A Declarative Approach for Flexible Business Processes Management. In J. Eder und S. Dustdar (Hg.), *Business Process Management Workshops*, Band 4103 von *LNCS*, S. 169–180. Springer Berlin Heidelberg (2006). ISBN 978-3-540-38444-1. doi:10.1007/11837862_18.

C. A. Petri. *Kommunikation mit Automaten*. Doktorarbeit, Universität Hamburg (1962).

J. B. Phipps. Dendrogram Topology. *Systematic Biology*, 20(3), S. 306 (1971). doi:10.2307/2412343. URL `http://dx.doi.org/10.2307/2412343`.

A. Polyvyanyy, S. Smirnov und M. Weske. On Application of Structural Decomposition for Process Model Abstraction. In *BPSC*, S. 110–122. Citeseer (2009).

A. Polyvyanyy, J. Vanhatalo und H. Völzer. *Simplified Computation and Generalization of the Refined Process Structure Tree*, S. 25–41. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). ISBN 978-3-642-19589-1. doi:10.1007/978-3-642-19589-1_2. URL `http://dx.doi.org/10.1007/978-3-642-19589-1_2`.

A. Polyvyanyy, M. Weidlich und M. Weske. Isotactics as a Foundation for Alignment and Abstraction of Behavioral Models. In A. Barros, A. Gal und E. Kindler (Hg.), *Business Process Management*, Band 7481 von *LNCS*, S. 335–351. Springer Berlin Heidelberg (2012). ISBN 978-3-642-32884-8.

M. Porter. An algorithm for suffix stripping. *Program*, 14(3), S. 130–137 (1980). doi:10.1108/eb046814. URL `http://dx.doi.org/10.1108/eb046814`.

J. Prescher, C. Di Ciccio und J. Mendling. From Declarative Processes to Imperative Models. In *SIMPDA*, S. 162–173 (2014).

M. Rana, K. Shahzad, R. M. A. Nawab, H. Leopold und U. Babar. *A Textual Description Based Approach to Process Matching*, S. 194–208. Springer International Publishing, Cham (2016). ISBN 978-3-319-48393-1. doi:10.1007/978-3-319-48393-1_14. URL http://dx.doi.org/10.1007/978-3-319-48393-1_14.

M. Reichert und B. Weber. *Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies*. Springer Berlin Heidelberg (2012). ISBN 978-3-642-30408-8.

H. A. Reijers und J. Mendling. A Study Into the Factors That Influence the Understandability of Business Process Models. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 41(3), S. 449–462 (2011). ISSN 1083-4427. doi:10.1109/TSMCA.2010.2087017.

M. M. Richter. *Classification and Learning of Similarity Measures*, S. 323–334. Springer Berlin Heidelberg, Berlin, Heidelberg (1993). ISBN 978-3-642-50974-2. doi:10.1007/978-3-642-50974-2_33. URL http://dx.doi.org/10.1007/978-3-642-50974-2_33.

C. Rodríguez, C. Klinkmüller, I. Weber, F. Daniel und F. Casati. *Activity Matching with Human Intelligence*, S. 124–140. Springer International Publishing, Cham (2016). ISBN 978-3-319-45468-9. doi:10.1007/978-3-319-45468-9_8. URL http://dx.doi.org/10.1007/978-3-319-45468-9_8.

M. L. Rosa, M. Dumas, C. C. Ekanayake, L. García-Bañuelos, J. Recker und A. H. ter Hofstede. Detecting approximate clones in business process model repositories. *Information Systems*, 49, S. 102 – 125 (2015). ISSN 0306-4379. doi:http://dx.doi.org/10.1016/j.is.2014.11.010. URL http://www.sciencedirect.com/science/article/pii/S0306437914001860.

M. Rosemann. Potential pitfalls of process modeling: Part A. *Business Process Management Journal*, 12(2), S. 249–254 (2006). ISSN 1463-7154. doi:10.1108/14637150610657567.

A. Rozinat und W. M. P. van der Aalst. *Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models*, S. 163–176. LNBIP. Springer Berlin Heidelberg (2006). ISBN 978-3-540-32596-3.

D. Sánchez-Charles, V. Muntés-Mulero, J. Carmona und M. Solé. *Process Model Comparison Based on Cophenetic Distance*, S. 141–158. Springer International Publishing, Cham (2016). ISBN 978-3-319-45468-9. doi:10.1007/978-3-319-45468-9_9. URL http://dx.doi.org/10.1007/978-3-319-45468-9_9.

A.-W. Scheer. *ARIS – Vom Geschäftsprozess zum Anwendungssystem*. Springer-Verlag Berlin Heidelberg (2002). ISBN 978-3-540-65823-8.

A. Schoknecht, T. Thaler, P. Fettke, A. Oberweis und R. Laue. Similarity of Business Process Models - A State-of-the-Art Analysis. *ACM Comput. Surv.*, 50(4), S. 52:1–52:33 (2017). ISSN 0360-0300. doi:10.1145/3092694. URL http://doi.acm.org/10.1145/3092694.

B. Schölkopf. The kernel trick for distances. In *Advances in neural information processing systems*, S. 301–307 (2001).

S. Schönig. *Ein Process Mining-Rahmenwerk für agile, personenbezogene Prozesse*. Doktorarbeit, Universität Bayreuth (2015).

S. Schönig, M. Zeising und S. Jablonski. *Towards Location-Aware Declarative Business Process Management*, S. 40–51. Springer International Publishing, Cham (2014). ISBN 978-3-319-11460-6. doi:10.1007/978-3-319-11460-6_4. URL http://dx.doi.org/10.1007/978-3-319-11460-6_4.

M. Sebag und M. Schoenauer. *A rule-based similarity measure*, S. 119–131. Springer Berlin Heidelberg (1994). ISBN 978-3-540-48655-8.

L. G. Shapiro und R. M. Haralick. Structural Descriptions and Inexact Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(5), S. 504–519 (1981). ISSN 0162-8828. doi:10.1109/TPAMI.1981.4767144.

H. Stachowiak. *Allgemeine Modelltheorie*. Springer-Verlag, Wien (1973).

J. Starlinger, B. Brancotte, S. Cohen-Boulakia und U. Leser. Similarity Search for Scientific Workflows. *Proc. VLDB Endow.*, 7(12), S. 1143–1154 (2014). ISSN 2150-8097.

T. Thaler, A. Schoknecht, P. Fettke, A. Oberweis und R. Laue. *A Comparative Analysis of Business Process Model Similarity Measures*, S. 310–322. Springer International Publishing, Cham (2017). ISBN 978-3-319-58457-7. doi:10.1007/978-3-319-58457-7_23. URL http://dx.doi.org/10.1007/978-3-319-58457-7_23.

M. Tka und S. A. Ghannouchi. Comparison of Business Process Models as Part of BPR Projects. *Procedia Technology*, 5, S. 427 – 436 (2012). ISSN 2212-0173. doi:http://dx.doi.org/10.1016/j.protcy.2012.09.047.

A. Tversky. Features of similarity. *Psychological Review*, 84(4), S. 327–352 (1977). doi:http://dx.doi.org/10.1037/0033-295X.84.4.327.

J. R. Ullmann. An Algorithm for Subgraph Isomorphism. *J. ACM*, 23(1), S. 31–42 (1976). ISSN 0004-5411. doi:10.1145/321921.321925. URL http://doi.acm.org/10.1145/321921.321925.

W. van der Aalst, A. ter Hofstede, B. Kiepuszewski und A. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1), S. 5–51 (2003a). ISSN 1573-7578. doi:10.1023/A:1022883727209. URL http://dx.doi.org/10.1023/A:1022883727209.

W. van der Aalst, T. Weijters und L. Maruster. Workflow mining: discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9), S. 1128–1142 (2004). ISSN 1041-4347. doi:10.1109/TKDE.2004.47.

W. M. van der Aalst und C. Stahl. *Modeling business processes: a petri net-oriented approach*. MIT press (2011).

W. M. P. van der Aalst, A. H. M. ter Hofstede und M. Weske. *Business Process Management: A Survey*, S. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg (2003b). ISBN 978-3-540-44895-2. doi:10.1007/3-540-44895-0_1. URL http://dx.doi.org/10.1007/3-540-44895-0_1.

B. van Dongen, R. Dijkman und J. Mendling. *Measuring Similarity between Business Process Models*, S. 405–419. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). ISBN 978-3-642-36926-1. doi:10.1007/978-3-642-36926-1_33. URL http://dx.doi.org/10.1007/978-3-642-36926-1_33.

B. f. van Dongen, J. Mendling und W. M. P. van der Aalst. Structural Patterns for Soundness of Business Process Models. In *2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*, S. 116–128 (2006). ISSN 1541-7719. doi:10.1109/EDOC.2006.56.

J. Vanhatalo, H. Völzer und J. Koehler. *The Refined Process Structure Tree*, S. 100–115. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). ISBN 978-3-540-85758-7. doi:10.1007/978-3-540-85758-7_10. URL http://dx.doi.org/10.1007/978-3-540-85758-7_10.

W3C OWL Working Group. OWL 2 Web Ontology Language (2012). URL https://www.w3.org/2012/pdf/REC-owl2-overview-20121211.pdf[accessed:2017-03-20].

B. Weber, M. Reichert, J. Mendling und H. A. Reijers. Refactoring large process model repositories. *Computers in Industry*, 62(5), S. 467 – 486 (2011). ISSN 0166-3615. doi:http://dx.doi.org/10.1016/j.compind.2010.12.012. URL http://www.sciencedirect.com/science/article/pii/S0166361510001843.

M. Weidlich, A. Barros, J. Mendling und M. Weske. *Vertical Alignment of Process Models – How Can We Get There?*, S. 71–84. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). ISBN 978-3-642-01862-6. doi:10.1007/978-3-642-01862-6_7. URL http://dx.doi.org/10.1007/978-3-642-01862-6_7.

M. Weidlich, G. Decker, M. Weske und A. Barros. Towards vertical alignment of process models-a collection of mismatches. *Hasso Plattner Institute, Tech. Rep* (2008).

M. Weidlich, R. Dijkman und J. Mendling. The ICoP Framework: Identification of Correspondences between Process Models. In B. Pernici (Hg.), *Advanced Information Systems Engineering*, Band 6051 von *LNCS*, S. 483–498. Springer Berlin Heidelberg (2010a). ISBN 978-3-642-13093-9.

M. Weidlich, A. Polyvyanyy, J. Mendling und M. Weske. Efficient Computation of Causal Behavioural Profiles Using Structural Decomposition. In J. Lilius und W. Penczek (Hg.), *Applications and Theory of Petri Nets: 31st International Conference, PETRI NETS 2010, Braga, Portugal, June 21-25, 2010. Proceedings*, S. 63–83. Springer Berlin Heidelberg (2010b). ISBN 978-3-642-13675-7.

M. Weske. *Business process management: concepts, languages, architectures*. Springer Publishing Company, Incorporated (2010).

W. E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, S. 354–359 (1990).

A. Wombacher. Evaluation of Technical Measures for Workflow Similarity Based on a Pilot Study. In R. Meersman und Z. Tari (Hg.), *On the Move to Meaningful Internet Systems (OTM)*, Band 4275 von *LNCS*, S. 255–272. Springer Berlin Heidelberg (2006). ISBN 978-3-540-48287-1.

A. Wombacher und M. Rozie. Evaluation of workflow similarity measures in service discovery. In *Service-Oriented Electronic Commerce: Proceedings zur Konferenz im Rahmen der Multikonferenz Wirtschaftsinformatik 2006*, S. 57–71. Gesellschaft fuer Informatik (2006).

J. Xing, X. Zhang, W. Song, Q. Yang, J. Ge und H. Wang. BPEL Similarity – A Metric Based on Activity Constraint Graphs. In M. Song, M. T. Wynn und J. Liu (Hg.), *AP-BPM 2013. Selected Papers*, S. 39–55. Springer International Publishing (2013). ISBN 978-3-319-02922-1.

Z. Yan, R. Dijkman und P. Grefen. *Fast Business Process Similarity Search with Feature-Based Similarity Estimation*, S. 60–77. Springer Berlin Heidelberg, Berlin, Heidelberg (2010). ISBN 978-3-642-16934-2. doi:10.1007/978-3-642-16934-2_8. URL `http://dx.doi.org/10.1007/978-3-642-16934-2_8`.

M. Zeising, S. Schönig und S. Jablonski. Towards a common platform for the support of routine and agile business processes. In *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, S. 94–103 (2014). doi:10.4108/icst.collaboratecom.2014.257269.

H. Zha, J. Wang, L. Wen, C. Wang und J. Sun. A workflow net similarity measure based on transition adjacency relations. *Computers in Industry*, 61(5), S. 463 – 471 (2010). ISSN 0166-3615. doi:https://doi.org/10.1016/j.compind.2010.01.001. URL `http://www.sciencedirect.com/science/article/pii/S0166361510000023`.

X. Zhao und C. Liu. *Version Management in the Business Process Change Context*, S. 198–213. Springer Berlin Heidelberg, Berlin, Heidelberg (2007). ISBN 978-3-540-75183-0. doi:10.1007/978-3-540-75183-0_15. URL `http://dx.doi.org/10.1007/978-3-540-75183-0_15`.

M. zur Muehlen und J. Recker. *How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation*, S. 429–443. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). ISBN 978-3-642-36926-1. doi:10.1007/978-3-642-36926-1_35. URL `https://doi.org/10.1007/978-3-642-36926-1_35`.

Abbildungsverzeichnis

Tabellenverzeichnis

Listingsverzeichnis

Eigene Publikationen

M. Bankau, M. Baumann, M. H. Baumann, S. Schönig und S. Jablonski. The Process Checklist Generator : Establishing Paper-based Process Support. In *Proceedings of the BPM Demo Track and BPM Dissertation Award* (2017). ISSN 1613-0073. URL `http://ceur-ws.org/Vol-1920/BPM_2017_paper_156.pdf`.

M. Baumann. Comparing Imperative and Declarative Process Models (2017). URL `https://epub.uni-bayreuth.de/3325/`. Preprint.

M. Baumann, M. H. Baumann, L. Ackermann, S. Schönig und S. Jablonski. Ansätze zum Ähnlichkeitsabgleich von deklarativen Geschäftsprozessmodellen. In *INFORMATIK 2016*, Band 259 von *Proceedings / GI-Edition*, S. 733–738. Köllen, Bonn (2016a). URL `http://subs.emis.de/LNI/Proceedings/Proceedings259/733.pdf`.

M. Baumann, M. H. Baumann, D. F.-X. Gruber und S. Jablonski. Infinite Horizon Decision Support For Rule-based Process Models. *International Journal on Advances in Software*, 9(1-2), S. 141–153 (2016b). URL `http://www.iariajournals.org/software/soft%5fv9%5fn12%5f2016%5fpaged.pdf`.

M. Baumann, M. H. Baumann und S. Jablonski. An Idea On Infinite Horizon Decision Support For Rule-based Process Models. In H. Kaindl, K. György und D. Tamir (Hg.), *ICCGI 2015, The Tenth International Multi-Conference on Computing in the Global Information Technology*, Band 5, S. 73–75. Think Mind (2015a). ISBN 978-1-61208-432-9.

M. Baumann, M. H. Baumann und S. Jablonski. On Behavioral Process Model Similarity Matching: A Centroid-based Approach. In *ICCGI 2015, The Tenth International Multi-Conference on Computing in the Global Information Technology*, Band 5, S. 125–131 (2015b).

M. Baumann, M. H. Baumann und S. Jablonski. On Behavioral Process Model Similarity Matching: A Centroid-based Approach (Enlarged Abstract of [BBJ15]). In *INFORMATIK 2016*, Band 259 von *Proceedings / GI-Edition*, S. 731–732. Köllen, Bonn (2016c). URL `http://subs.emis.de/LNI/Proceedings/Proceedings259/731.pdf`.

M. Baumann, M. H. Baumann, S. Schönig und S. Jablonski. Enhancing Feasibility of Human-Driven Processes by Transforming Process Models to Process Checklists. In I. Bider, K. Gaaloul, J. Krogstie, S. Nurcan, H. A. Proper, R. Schmidt und P. Soffer (Hg.), *Enterprise, Business-Process and Information Systems Modeling*, Band 175 von *LNBIP*, S. 124–138. Springer Berlin Heidelberg (2014a). ISBN 978-3-662-43744-5. doi: 10.1007/978-3-662-43745-2_9.

M. Baumann, M. H. Baumann, S. Schönig und S. Jablonski. Resource-Aware Process Model Similarity Matching. In F. Toumani und et al. (Hg.), *Service-Oriented Computing - ICSOC 2014 Workshops*, Band 8954 von *Lecture Notes in Computer Science*, S. 96–107. Springer International Publishing (2015c). ISBN 978-3-319-22884-6. doi: 10.1007/978-3-319-22885-3_9.

M. Baumann, M. H. Baumann, S. Schönig und S. Jablonski. The Process Checklist. *Enterprise Modelling and Information Systems Architectures*, 12, S. 1–42 (2017). doi:http://dx.doi.org/10.18417/emisa.12.1.

M. H. Baumann, M. Baumann, S. Schönig und S. Jablonski. Towards Multi-perspective Process Model Similarity Matching. In J. Barjis und R. Pergl (Hg.), *Enterprise and Organizational Modeling and Simulation*, Band 191 von *LNBIP*, S. 21–37. Springer Berlin Heidelberg (2014b). ISBN 978-3-662-44859-5.

Danksagung

Beim Verfassen dieser Arbeit wurde ich von vielen lieben Menschen unterstützt, bei denen ich mich an dieser Stelle recht herzlich bedanke.

In erster Linie geht mein Dank an meinen Doktorvater, Prof. Dr.-Ing. Stefan Jablonski, der mir die Chance zur Promotion gegeben hat. Seine Hinweise und Anregungen sowie seine Aufgeschlossenheit gegenüber neuen Ideen haben diese Arbeit erst möglich gemacht.

Ebenso bedanke ich mich bei meinen Koautoren und Kollegen am Lehrstuhl Angewandte Informatik IV und an befreundeten Lehrstühlen für die gute Zusammenarbeit in den vergangenen Jahren, insbesondere bei Michael Heinrich Baumann, Prof. Dr. Bernhard Herz, Dr. Stefan Schönig und Lars Ackermann sowie bei Marcel Bankau und Susanne Hoffmeister.

Nicht unerwähnt lassen möchte ich die zahlreichen Editoren, Reviewer, Leser und Zuhörer meiner Arbeiten bzw. Konferenzbeiträge, für deren wertvolle Tipps ich ihnen sehr verbunden bin. Besonders den Veranstaltern und Besuchern des Forschungskolloquiums Modellierung, welches von zahlreichen Lehrstühlen verschiedenster Universitäten regelmäßig veranstaltet wird, und des Forschungszentrums Modellierung und Simulation (MODUS) der Universität Bayreuth möchte ich meinen Dank für die vielen wertvollen Diskussionen aussprechen.

Zu guter Letzt bedanke ich mich bei meiner Familie und meinen Freunden, insbesondere meinem Papa Bernhard, meiner Schwiegermama Monika, meinen Schwestern Martina und Franziska sowie meinem Mann Michael, die jederzeit ein offenes Ohr für mich haben.